Europäisches Patentamt	European Patent Office	Office européen des brevets

Bescheinigung **Certificate** **Attestation**

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.	The attached documents are exact copies of the European patent application described on the following page, as originally filed.	Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

00400840.5

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,   26/06/00
LA HAYE, LE

THIS PAGE BLANK (USPTO)

Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:   00400840.5
Demande n°:

Anmeldetag:
Date of filing: 27/03/00
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.

5621 BA Eindhoven

NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
   Architecture for the transport of mpeg-4 data over mpeg-2 system

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

«ARCHITECTURE FOR THE TRANSPORT OF MPEG-4 DATA OVER MPEG-2 SYSTEM"

FIELD OF THE INVENTION

5   The present invention relates to a global end-to-end system architecture for the transport of MPEG-4 data over MPEG-2 Systems. It also relates, in such an architecture, to a server platform receiving an MPEG-2 transport stream and MPEG-4 data to be incorporated within said stream and to encapsulating devices for carrying out the encapsulation of said MPEG-4 data in this existing MPEG-2 transport stream.

10   **BACKGROUND OF THE INVENTION**

The document ISO/IEC-JTC1/SC29/WG11/N3050 ("Information technology - Generic coding of moving pictures and audio : systems", Amendment 7 : Transport of ISO/IEC-14496 data over ISO/IEC-13818-1, January 2000) is considered as the technological background of the present invention. This document is enclosed (pp.2-31).

15   **DESCRIPTION OF THE INVENTION**

The present invention is described in pp.32-45 and claimed in p.46.

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 **N3050**

January 2000

Systems
Title: **ISO/IEC 13818-1/FDAM 7**

INFORMATION TECHNOLOGY –

GENERIC CODING OF MOVING PICTURES AND AUDIO: SYSTEMS

Amendment 7: Transport of ISO/IEC 14496 data over ISO/IEC 13818-1

ISO/IEC 13818-1/Final Draft Amendment 7

International Standard

ISO/IEC 13818-1:1996/Amd.7:199x(E)

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY -- GENERIC CODING OF MOVING
PICTURES AND ASSOCIATED AUDIO INFORMATION: SYSTEMS

AMENDMENT 7

1) Replace table 2-18 in subclause 2:

"

Table 2-18 — Stream_id assignments

stream_id	Note	stream coding
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream_2
110x xxxx		ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx		ITU-T Rec. H.262 | ISO/IEC 13818-2 or ISO/IEC 11172-2 or ISO/IEC 14496-2 video stream number xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Annex A or ISO/IEC 13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1 type A
1111 0101	6	ITU-T Rec. H.222.1 type B
1111 0110	6	ITU-T Rec. H.222.1 type C
1111 0111	6	ITU-T Rec. H.222.1 type D
1111 1000	6	ITU-T Rec. H.222.1 type E
1111 1001	7	ancillary_stream
1111 1010		ISO/IEC14496-1_SL-packetized_stream
1111 1011		ISO/IEC14496-1_FlexMux_stream
1111 1100 ... 1111 1110		reserved data stream
1111 1111	4	program_stream_directory

ITU-T Rec. H.222.0 (1996)/Amd.7(199x E)

The notation x means that the values '0' or '1' are both permitted and results in the same stream type. The stream number is given by the values taken by the x's.

NOTES

1   PES packets of type program_stream_map have unique syntax specified in 2.5.4.1.

2   PES packets of type private_stream_1 and ISO/IEC_13552_stream follow the same PES packet syntax as those for ITU-T Rec. H.262 | ISO/IEC 13818-2 video and ISO/IEC 13818-3 audio streams.

3   PES packets of type private_stream_2, ECM_stream and EMM_stream are similar to private_stream_1 except no syntax is specified after PES_packet_length field.

4   PES packets of type program_stream_directory have a unique syntax specified in 2.5.5.

5   PES packets of type DSM-CC_stream have a unique syntax specified in ISO/IEC 13818- 6.

6   This stream_id is associated with stream_type 0x09 in Table 2-29.

7   This stream_id is only used in PES packets, which carry data from a Program Stream or an ISO/IEC 11172-1 System Stream, in a Transport Stream (refer to 2.4.3.7).

2) Replace table 2-26 in subclause 2:

"

Table 2-26 — table_id assignment values

value	description
0x00	program association section
0x01	conditional access section (CA section)
0x02	TS program map section
0x03	TS description section
0x04	ISO IEC 14496 scene description section
0x05	ISO IEC 14496 object descriptor section
0x06–0x37	ITU-T Rec. H.222.0	ISO/IEC 13818-1 reserved
0x38–0x3F	Defined in ISO/IEC 13818-6
0x40–0xFE	User private
0xFF	forbidden

"

3) Replace table 2-29 in subclause 2:

"

Table 2-29 — Stream type assignments

Value	Description
0x00	ITU-T	ISO/IEC Reserved
0x01	ISO/IEC 11172 Video
0x02	ITU-T Rec. H.262	ISO/IEC 13818-2 Video or ISO/IEC 11172-2 constrained parameter video stream
0x03	ISO/IEC 11172 Audio
0x04	ISO/IEC 13818-3 Audio
0x05	ITU-T Rec. H.222.0	ISO/IEC 13818-1 private_sections
0x06	ITU-T Rec. H.222.0	ISO/IEC 13818-1 PES packets containing private data
0x07	ISO/IEC 13522 MHEG
0x08	ITU-T Rec. H.222.0	ISO/IEC 13818-1 Annex A DSM CC

4

ISO/IEC 13818-1:1996/Amd.7:199x(E)

0x09	ITU-T Rec. H.222.1
0x0A	ISO/IEC 13818-6 type A
0x0B	ISO/IEC 13818-6 type B
0x0C	ISO/IEC 13818-6 type C
0x0D	ISO/IEC 13818-6 type D
0x0E	ITU-T Rec. H.222.0	ISO/IEC 13818-1 auxiliary
0x0F	ISO/IEC 13818-7 Audio with ADTS transport syntax
0x10	ISO/IEC 14496-2 Visual
0x11	ISO/IEC 14496-3 Audio with the LATM transport syntax as defined in ISO/IEC 14496-3 / AMD 1
0x12	ISO/IEC 14496-1 SL-packetized stream or FlexMux stream carried in PES packets
0x13	ISO/IEC 14496-1 SL-packetized stream or FlexMux stream carried in ISO/IEC14496 sections.
0x14	ISO/IEC 13818-6 Synchronized Download Protocol
0x15-0x7F	ITU-T Rec. H.222.0	ISO/IEC 13818-1 Reserved
0x80-0xFF	User Private

4) Replace table 2-39 in subclause 2:

"

Table 2-39 — Program and program element descriptors

descriptor_tag	TS	PS	Identification
0	n/a	n/a	Reserved
1	n/a	n/a	Reserved
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	Video_window_descriptor
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	System_clock_descriptor
12	X	X	Multiplex_buffer_utilization_descriptor
13	X	X	Copyright_descriptor
14	X		Maximum_bitrate_descriptor
15	X	X	Private_data_indicator_descriptor
16	X	X	Smoothing_buffer_descriptor
17	X		STD_descriptor
18	X	X	IBP_descriptor
19-26	X		Defined in ISO/IEC 13818-6
27	X	X	MPEG-4_video_descriptor
28	X	X	MPEG-4_audio_descriptor
29	X	X	IOD_descriptor
30	X		SL_descriptor
31	X	X	FMC_descriptor
32	X	X	External_ES_ID_descriptor
33	X	X	MuxCode_descriptor
34	X	X	FmxBufferSize_descriptor
35	X		MultiplexBuffer_descriptor
36-63	n/a	n/a	ITU-T Rec. H.222.0	ISO/IEC 13818-1 Reserved

ITU-T Rec. H.222.0 (1996)/Amd.7(199x E)

64-255	n/a	n/a	User Private

5) Add after subclause 2.6.35:

"

2.6.36 MPEG-4 video descriptor

For individual ISO/IEC 14496-2 streams directly carried in PES packets, as defined in clause 2.11.2, the MPEG-4 video descriptor provides basic information for identifying the coding parameters of such visual elementary streams. The MPEG-4 video descriptor does not apply to ISO/IEC 14496-2 streams encapsulated in SL-packets and in FlexMux packets, as defined in clause 2.11.3.

Syntax	No. of bits	Mnemonic
MPEG-4_video_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MPEG-4_visual_profile_and_level	8	uimsbf
}		

2.6.37 Semantic definition of fields in MPEG-4 video descriptor

MPEG-4_video_profile_and_level - This 8-bit field shall identify the profile and level of the ISO/IEC 14496-2 video stream. This field shall be coded with the same value as the profile_and_level_indication field in the Visual Object Sequence Header in the associated ISO/IEC 14496-2 stream.

2.6.38 MPEG-4 audio descriptor

For individual ISO/IEC 14496-3 streams directly carried in PES packets, as defined in clause 2.11.2, the MPEG-4 audio descriptor provides basic information for identifying the coding parameters of such audio elementary streams. The MPEG-4 audio descriptor does not apply to ISO/IEC 14496-3 streams encapsulated in SL-packets and in FlexMux packets, as defined in clause 2.11.3.

Syntax	No. of bits	Mnemonic
MPEG-4_audio_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MPEG-4_audio_profile_and_level	8	uimsbf
}		

2.6.39 Semantic definition of fields in MPEG-4 audio descriptor

MPEG-4_audio_profile_and_level - This 8-bit field shall identify the profile and level of the ISO/IEC 14496-3 audio stream corresponding to the following table.

Table 2-27 – MPEG-4_audio_profile_and_level assignment values

Value	Description

ISO/IEC 13818-1:1996/Amd.7:199x(E)

0x00 - 0x0F	Reserved
0x10	Main profile, level 1
0x11	Main profile, level 2
0x12	Main profile, level 3
0x13	Main profile, level 4
0x14 – 0x17	Reserved
0x18	Scalable Profile, level 1
0x19	Scalable Profile, level 2
0x1A	Scalable Profile, level 3
0x1B	Scalable Profile, level 4
0x1C – 0x1F	Reserved
0x20	Speech profile, level 1
0x21	Speech profile, level 2
0x22 – 0x27	Reserved
0x28	Synthesis profile, level 1
0x29	Synthesis profile, level 2
0x2A	Synthesis profile, level 3
0x2B – 0x2F	Reserved
0x30	High quality audio profile, level 1
0x31	High quality audio profile, level 2
0x32	High quality audio profile, level 3
0x33	High quality audio profile, level 4
0x34 – 0x37	Reserved
0x38	Low delay audio profile, level 1
0x39	Low delay audio profile, level 2
0x3A	Low delay audio profile, level 3
0x3B	Low delay audio profile, level 4
0x3C – 0x3F	Reserved
0x40	Natural audio profile, level 1
0x41	Natural audio profile, level 2
0x42 – 0x47	Reserved
0x48	Mobile communications profile, level 1
0x49	Mobile communications profile, level 2
0x4A	Mobile communications profile, level 3
0x4B – 0xFF	Reserved

2.6.40 IOD descriptor

The IOD descriptor encapsulates the InitialObjectDescriptor structure. An initial object descriptor allows access to a set of ISO/IEC 14496 streams by identifying the ES_ID values of the ISO/IEC 14496-1 scene description and object descriptor streams. Both the scene description stream and the object descriptor stream contain further information about the ISO/IEC 14496 streams that are part of the scene. See Annex R for a description of the content access procedure. The InitialObjectDescriptor is specified in sub-clause 8.6.3 of ISO/IEC 14496-1.

Within a Transport Stream, the IOD descriptor shall be conveyed in the descriptor loop immediately following the program_info_length field in the Program Map Table. If a Program Stream Map is present in

a Program Stream, the IOD descriptor shall be conveyed in the descriptor loop immediately following the program_stream_info_length field in the Program Stream Map. More than one IOD descriptor may be associated to a program.

NOTE - This specification does not specify how the IOD_label may be used by higher level service information to uniquely select one of the ISO/IEC 14496 presentations identified by multiple IOD descriptors.

Syntax	No. of bits	Mnemonic
IOD_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Scope_of_IOD_label	8	uimsbf
IOD_label	8	uimsbf
InitialObjectDescriptor ()		
}		

2.6.41 Semantic definition of fields in IOD descriptor

Scope_of_IOD_label - This 8-bit field specifies the scope of the IOD_label field. A value of 0x10 indicates that the IOD_label is unique within the Program Stream or within the specific program in a Transport Stream in which the IOD descriptor is carried. A value of 0x11 indicates that the IOD_label is unique within the Transport Stream in which the IOD descriptor is carried. All other values of the Scope_of_IOD_label field are reserved.

IOD_label - This 8-bit field specifies the label of the IOD descriptor.

InitialObjectDescriptor () - This structure is defined in sub-clause 8.6.3.1 of ISO/IEC 14496-1.

2.6.42 SL descriptor

The SL descriptor shall be used when a single ISO/IEC 14496-1 SL-packetized stream is encapsulated in PES packets. The SL descriptor associates the ES_ID of this SL-packetized stream to an elementary_PID in case of a Transport Stream or to an elementary_stream_id in case of a Program Stream. Within a Transport Stream, the SL descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the SL descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field within the Program Stream Map.

NOTE – SL packetized streams may be used in a Program Stream. However, only one stream_id exists for ISO/IEC 14496-1 SL-packetized streams. In order to associate multiple such streams within a Program Stream to an ISO/IEC 14496-1 scene, FlexMux has to be used and signaled appropriately by an FMC descriptor. This limitation does not exist in a Transport Stream where the SL descriptor provides unambiguous mapping between an ISO/IEC 14496-1 ES_ID value and an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 elementary_PID value.

Syntax	No. of bits	Mnemonic
SL_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
ES_ID	16	uimsbf
}		

Syntax	No. of bits	Mnemonic
External_ES_ID_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
External_ES_ID	16	uimsbf
}		

2.6.47 Semantic definition of fields in External_ES_ID descriptor

External_ES_ID - This 16-bit field assigns an ES_ID identifier, as defined in ISO/IEC 14496-1, to a component of a program.

2.6.48 Muxcode descriptor

The Muxcode descriptor conveys MuxCodeTableEntry structures as defined in subclause 11.2.4.3 of ISO/IEC 14496-1. MuxCodeTableEntries configure the MuxCode mode of FlexMux.

One or more Muxcode descriptors may be associated to each elementary_PID or elementary_stream_id, respectively, conveying an ISO/IEC 14496-1 FlexMux stream that utilizes the MuxCode mode. Within a Transport stream, the muxcode descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the Muxcode descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field in the Program Stream Map.

MuxCodeTableEntries may be updated with new versions. In case of such updates, the version_number of each Program Map Table or the program_stream_map_version of each Program Stream Map, respectively, carrying the MuxCode descriptor in their descriptor loop shall be incremented by 1 modulo 32.

Syntax	No. of bits	Mnemonic
Muxcode_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i = 0; i < N; i++) {		
MuxCodeTableEntry ()		
}		
}		

2.6.49 Semantics of fields in Muxcode descriptor

MuxCodeTableEntry () - This structure is defined in sub-clause 11.2.4.3 of ISO/IEC 14496-1.

2.6.50 FmxBufferSize descriptor

The FmxBufferSize descriptor conveys the size of the FlexMux buffer (FB) for each SL packetized stream multiplexed in a FlexMux stream.

One FmxBufferSize descriptor shall be associated to each elementary_PID or elementary_stream_id, respectively, conveying an ISO/IEC 14496-1 FlexMux stream. Within a Transport stream, the FmxBufferSize descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the FmxBufferSize descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field within the Program Stream Map.

2.6.43 Semantic definition of fields in SL descriptor

ES_ID - This 16-bit field shall specify the identifier of an ISO/IEC 14496-1 SL-packetized stream.

2.6.44 FMC descriptor

The FMC descriptor indicates that the ISO/IEC 14496-1 FlexMux tool has been used to multiplex ISO/IEC 14496-1 SL-packetized streams into a FlexMux stream before encapsulation in PES packets or ISO//IEC14496_sections. The FMC descriptor associates FlexMux channels to the ES_ID values of the SL-packetized streams in the FlexMux stream.

An FMC descriptor is required for each program element referenced by an elementary_PID value in a Transport Stream and for each elementary_stream_id in a Program Stream that conveys a FlexMux stream. Within a Transport Stream, the FMC descriptor shall be conveyed for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the FMC descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field in the Program Stream Map.

For each SL_packetized stream in a FlexMux stream , the FlexMux channel shall be identified by a single entry in the FMC descriptor.

Syntax	No. of bits	Mnemonic
FMC_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<descriptor_length; i += 3) {		
ES_ID	16	uimsbf
FlexMuxChannel	8	uimsbf
}		
}		

2.6.45 Semantic definition of fields in FMC descriptor

ES_ID - This 16-bit field specifies the identifier of an ISO/IEC 14496-1 SL-packetized stream.

FlexMuxChannel - This 8-bit field specifies the number of the FlexMux channel used for this SL-packetized stream.

2.6.46 External_ES_ID descriptor

The External_ES_ID descriptor assigns an ES_ID, as defined in ISO/IEC 14496-1, to a program element to which no ES_ID value has been assigned by other means. This ES_ID allows reference to a non-ISO/IEC 14496 component in the scene description or, for example, to associate a non-ISO/IEC 14496 component with an IPMP stream.

Within a Transport stream, the assignment of an ES_ID shall be made by conveying an External_ES_ID descriptor for the corresponding elementary stream in the descriptor loop immediately following the ES_info_length field in the Program Map Table. If a Program Stream Map is present in a Program Stream, the External_ES_ID descriptor shall be conveyed in the descriptor loop immediately following the elementary_stream_info_length field in the Program Stream Map.

Syntax	No. of bits	Mnemonic

ISO/IEC 13818-1:1996/Amd.7:199x(E)

Syntax	No. of bits	Mnemonic
FmxBufferSize_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
DefaultFlexMuxBufferDescriptor()		
for (i=0; i<descriptor_length; i += 4) {		
FlexMuxBufferDescriptor()		
}		
}		

2.6.51 Semantics of fields in FmxBufferSize descriptor

FlexMuxBufferDescriptor() - This descriptor specifies the FlexMux buffer size for one SL-packetized stream carried within the FlexMux stream. It is defined in sub-clause 11.2 of ISO/IEC 14496-1.

DefaultFlexMuxBufferDescriptor() - This descriptor specifies the default FlexMux buffer size for this FlexMux stream. It is defined in sub-clause 11.2 of ISO/IEC 14496-1.

2.6.52 MultiplexBuffer descriptor

The MultiplexBuffer descriptor conveys the size of the multiplex buffer MB_n, as well as the leak rate Rx_n at which data is transferred from transport buffer TB_n into buffer MB_n for a specific ISO/IEC 13818-1 program element referenced by an elementary_PID value in the Program Map Table.

One MultiplexBuffer descriptor shall be associated to each elementary_PID that contains an ISO/IEC 14496 FlexMux stream or SL-packetized stream, including those containing ISO_IEC_14496_sections. See clause 2.11.3.9 for the definition of buffers and rates in the T-STD model for decoding of ISO/IEC 14496 content.

The MultiplexBuffer descriptor shall be conveyed in the descriptor loop immediately following the ES_info_length field in the Program Map Table.

Syntax	No. of bits	Mnemonic
MultiplexBuffer_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
MB_buffer_size	24	uimsbf
TB_leak_rate	24	uimsbf
}		

2.6.53 Semantics of fields in MultiplexBuffer descriptor

MB_buffer_size - This 24-bit field shall specify the size in byte of buffer MB_n of the elementary stream n that is associated with this descriptor.

TB_leak_rate - This 24-bit field shall specify in units of 400 bits per second the rate at which data is transferred from transport buffer TB_n to multiplex buffer MB_n for the elementary stream n that is associated with this descriptor.

"

6) Add after subclause 2.4.2.6:

2.4.2.7 T-STD extensions for carriage of ISO/IEC 14496 data

For decoding of ISO/IEC 14496 data carried in a Transport Stream the T-STD model is extended. T-STD parameters for decoding of individual ISO/IEC 14496 elementary streams are defined in clause 2.11.2, while clause 2.11.3 defines T-STD extensions and parameters for decoding of ISO/IEC 14496 scenes and associated streams.

7) *Add after subclause 2.5.2.5:*

2.5.2.6 P-STD extensions for carriage of ISO/IEC 14496 data

For decoding of ISO/IEC 14496 data carried in a Program Stream the P-STD model is extended. For decoding of individual ISO/IEC 14496 elementary streams in the P-STD see clause 2.11.2. Clause 2.11.3 defines P-STD extensions and parameters for decoding of ISO/IEC 14496 scenes and associated streams.

8) *Changes to clause 2.4.2.6*

Replace the text

> The delay of any data through the System Target Decoders buffers shall be less than or equal to one second except for still picture video data.

with

> The delay of any data through the System Target Decoder buffers shall be less than or equal to one second except for still picture video data and ISO/IEC 14496 streams.

Add the following paragraph immediately before the Definition of overflow and underflow :

> For ISO/IEC 14496 streams, the delay is constrained by $td_n(j)-t(i) <= 10$ second for all j, and all bytes i in access unit $A_n(j)$.

9) *Changes to clause 2.5.2.3*

Replace the text

> For all Program Streams, the delay caused by system target decoder input buffering shall be less than or equal to one second except for still picture video data.

with

> For all Program Streams, the delay caused by system target decoder input buffering shall be less than or equal to one second except for still picture video data and ISO/IEC 14496 streams.

Replace the text

> Specifically: in the case of no still picture video data the delay is constrained by:
> $td_n(j)-t(i) <= 1$ sec
> else in the case of still picture video data the delay is constrained by:
> $td_n(j)-t(i) <= 60$ sec
> for all bytes contained in access unit j.

with

Specifically: in the case of no still picture video data and no ISO/IEC 14496 stream the delay is constrained by:

$$td_n(j)-t(i) <= 1 \text{ sec}$$

in the case of still picture video data the delay is constrained by:

$$td_n(j)-t(i) <= 60 \text{ sec}$$

in the case of ISO/IEC 14496 streams the delay is constrained by:

$$td_n(j)-t(i) <= 10 \text{ sec}$$

for all bytes contained in access unit j.

10)Add clause 2.11 immediately after clause 2.10:

2.11 Carriage of ISO/IEC 14496 data

2.11.1 Introduction

An ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream may carry individual ISO/IEC 14496-2 and 14496-3 elementary streams as well as ISO/IEC 14496-1 audiovisual scenes with its associated streams. Typically, the ISO/IEC 14496 streams will be elements of an ISO/IEC 13818-1 program, as defined by the PMT in a Transport Stream and the PSM in a Program Stream.

For the carriage of ISO/IEC 14496 data in Transport Streams and Program Streams distinction is made between individual elementary streams and an ISO/IEC 14496-1 audiovisual scene with its associated streams.

- For carriage of individual ISO/IEC 14496-2 and 14496-3 elementary streams only system tools from ITU-T Rec. H.222.0 | ISO/IEC 13818-1 are used, as defined in clause 2.11.2;

- For carriage of an audiovisual ISO/IEC 14496-1 scene and associated ISO/IEC 14496 elementary streams, contained in ISO/IEC 14496-1 SL_packetized streams or FlexMux streams, tools from both ITU-T Rec. H.222.0 | ISO/IEC 13818-1 and from ISO/IEC 14496-1 are used, as defined in clause 2.11.3.

2.11.2 Carriage of individual ISO/IEC 14496-2 and 14496-3 Elementary Streams in PES packets

2.11.2.1   Introduction

Individual ISO/IEC 14496-2 and 14496-3 elementary streams may be carried in PES packets as PES_packet_data_bytes. For PES packetization no specific data alignment constraints apply. For synchronisation PTSs and, when appropriate, DTSs are encoded in the header of the PES packet that carries the ISO/IEC 14496 elementary stream data; for PTS and DTS encoding the same constraints apply as for ISO/IEC 13818 elementary streams. See table 2-62 for an overview of how to carry individual ISO/IEC 14496 streams within an ISO/IEC ITU-T Rec. H.222.0 | 13818-1 stream.

Table 2-62 – Carriage of individual ISO/IEC 14496 streams in ITU-T Rec. H.222.0 | ISO/IEC 13818-1

ISO/IEC 14496-2 visual	Carriage in PES packets	Stream_type = 0x10	Stream_id = '1110 xxxx'
ISO/IEC 14496-3 audio	Carriage in PES packets	Stream_type = 0x11	Stream_id = '110x xxxx'

If a PTS or DTS is present in the PES packet header it shall refer to the visual object that follows either the first VOP start code or the first still texture object startcode that commences in the PES packet. Each ISO/IEC 14496-2 video stream carried by ITU-T Rec. H.222.0 | ISO/IEC 13818-1 shall contain the

information required to decode the ISO/IEC 14496-2 video stream; consequently the stream shall contain Visual Object Sequence Headers, Visual Object Headers and Video Object Layer Headers.

In case of an ISO/IEC 14496-3 elementary stream, before PES packetization the elementary stream data shall be first encapsulated in the LATM transport syntax defined in ISO/IEC 14496-3 / AMD 1. If a PTS is present in the PES packet header it shall refer to the first audio frame that follows the first syncword that commences in the payload of the PES packet.

Carriage of individual ISO/IEC 14496 elementary streams in PES packets shall be identified by appropriate stream_id and stream_type values, indicating the use of ISO/IEC 14496-2 Visual or 14496-3 Audio. In addition, such carriage shall be signaled by the MPEG-4_video descriptor or MPEG-4_audio descriptor, respectively. These descriptors shall be conveyed in the descriptor loop for the respective elementary stream entry in the Program Map Table in case of a Transport Stream or in the Program Stream Map, when present, in case of a Program Stream. ITU-T Rec. H.222.0 | ISO/IEC 13818-1 does not specify presentation of ISO/IEC 14496 elementary streams in the context of a program.

2.11.2.2   STD extensions for individual ISO/IEC 14496 elementary streams

The T-STD model includes a transport buffer TB_n and a multiplex buffer B_n prior to decoding of each individual ISO/IEC 14496 elementary stream n. Note that in the T-STD the single multiplex buffer B_n is also applied for ISO/IEC 14496-2 video, as indicated in Figure 2-4, instead of the approach with two buffers MB_n and EB_n used for ISO/IEC 13818-2 video in the T-STD. For buffers TB_n and B_n and the rate Rx_n between TB_n and B_n the following constraints apply.
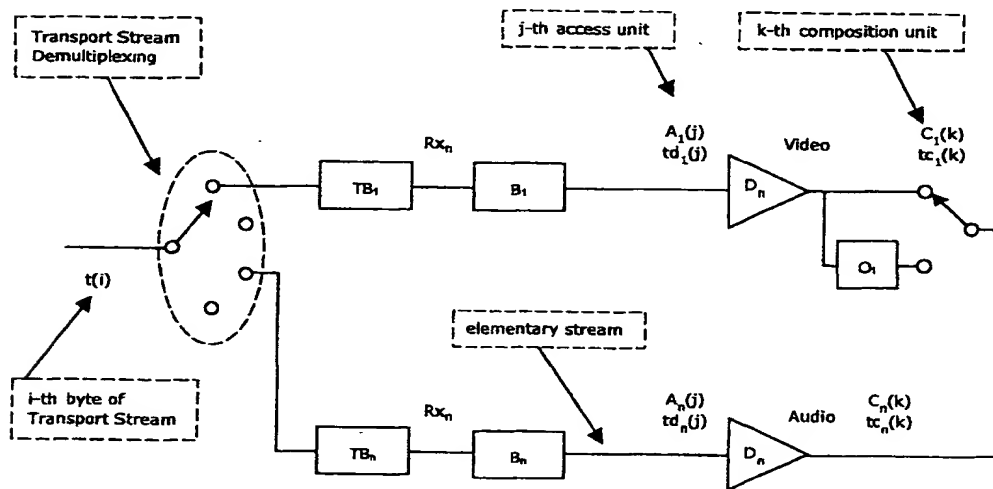
Figure 2-4 – T-STD model extensions for individual ISO/IEC 14496 elementary streams

In case of carriage of a ISO/IEC 14496-2 stream:

Size BS_n of Buffer B_n :

$$BS_n = BS_{mux} + BS_{oh} + VBV_{max}[profile, level]$$

where BS_{oh}, packet overhead buffering, is defined as:

$$BS_{oh} = (1/750) \text{ seconds} \times \max\{R_{max}[profile, level], 2\,000\,000 \text{ bit/second}\}$$

and BS_{mux}, additional multiplex buffering, is defined as:

$$BS_{mux} = 0.004 \text{ seconds} \times \max\{R_{max}[profile, level], 2\,000\,000 \text{ bit/second}\}$$

Rate Rx_n :

$$Rx_n = 1.2 \times R_{max}[profile,level]$$

Where $VBV_{max}[profile,level]$ and $R_{max}[profile,level]$ are defined in ISO/IEC 14496-2 for each profile and level. For profiles and levels for which no VBV_{max} value is specified, the size of B_n and the rate Rx_n are user defined.

In case of carriage of an ISO/IEC 14496-3 stream:

Size BS_n of Buffer B_n for ISO/IEC 14496-3 AAC audio as defined in this Specification,

else $BS_n = BS_{mux} + BS_{dec} + BS_{ob} = 3584$ bytes

In this case the size of the access unit decoding buffer BS_{dec}, and the PES packet overhead buffer BS_{ob} are constrained by

$$BS_{dec} + BS_{ob} \quad 2848 \text{ bytes}$$

A portion (736 bytes) of the 3584 byte buffer is allocated for buffering to allow multiplexing. The rest, 2848 bytes, are shared for access unit buffering BS_{dec}, BS_{ob} and additional multiplexing.

Rate Rx_n for ISO/IEC 14496-3 AAC audio as defined in this Specification,

else $Rx_n = 2\,000\,000$ bit/sec.

The P-STD model includes a multiplex buffer B_n prior to decoding of each individual ISO/IEC 14496 elementary stream n. The size BS_n of buffer B_n in the P-STD is defined by the P-STD_buffer_size field in the PES packet header.

2.11.3 Carriage of audiovisual ISO/IEC 14496-1 scenes and associated ISO/IEC 14496 streams

2.11.3.1   Introduction

This clause describes the encapsulation and signaling when an audiovisual scene represented by ISO/IEC 14496 data is carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream or Transport Stream. ISO/IEC 14496 content consists of the initial object descriptor and a variable number of streams such as object descriptor streams, scene description streams (carrying either BIFS-Command or BIFS-Anim access units), IPMP streams, OCI streams and audio-visual streams. Each of the ISO/IEC 14496 streams shall be contained in an SL-packetized stream and may optionally be multiplexed into a FlexMux stream, both defined in ISO/IEC 14496-1. For carriage in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream or Transport Stream, these SL-packetized streams and FlexMux streams shall contain encoded Object Clock Reference (OCR) and FlexMux Clock Reference (FCR) fields as specified in 2.11.3.4 and in 2.11.3.5, respectively. The SL-packetized streams or FlexMux streams are then encapsulated either in PES packets or in ISO_IEC_14496_sections prior to Transport Stream packetization and multiplexing or Program Stream multiplexing. ISO_IEC_14496_sections are built on the long format of H.222.0 | ISO/IEC 13818-1 sections.

2.11.3.2   Assignment of ES_ID values

An ISO/IEC 14496-1 scene carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream may associate a number of ISO/IEC 14496, ISO/IEC 13818 and other streams by the use of the ES_ID parameter. The scene and the associated streams may be carried over the same ITU-T Rec. H.222.0 | ISO/IEC 13818-1

stream, but a scene may also reference streams carried elsewhere, for example over an IP network. How to identify such other means is not defined in this Specification.

ISO/IEC 14496-1 defines name scoping rules for identifiers. These rules allow the same ES_ID value to be used for two different streams within ISO/IEC 14496 content. When one or multiple ISO/IEC 14496-1 scenes are carried in a ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program, duplicate ES_ID values shall not occur within the program such that each ISO/IEC 14496 SL-packetized stream or ISO/IEC 14496-1 Flexmux channel has a unique ES_ID value in the program.

2.11.3.3   Timing of ISO/IEC 14496 scenes and associated streams

When carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, the object time base of each ISO/IEC 14496 stream shall be locked to the ITU-T Rec. H.222.0 | ISO/IEC 13818-1 STC, that is :

If $X(t) = f_{stc}(t) / f_{object}(t)$, then

the value of $X(t)$ shall be constant at any time t

Where $f_{stc}(t)$ denotes the intended frequency of the STC at time t, i.e. 27 000 000 Hz

$f_{object}(t)$ denotes the frequency of the object time base at time t

The object time base of ISO/IEC 14496 streams carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream is conveyed as follows :

-    The object time base of an SL-packetized stream carried in PES packets without the use of the FlexMux shall be conveyed by coded OCRs in the SL packet header of that stream. See clause 2.11.3.4.
-    The object time base of SL-packetized streams carried in PES packets within a FlexMux stream shall be conveyed by FCRs in that FlexMux stream. See clause 2.11.3.5. Consequently, all ISO/IEC 14496 streams contained within the same FlexMux stream share the same object time base.
-    The object time base of an SL-packetized stream carried in sections shall be conveyed by another ISO/IEC 14496 stream within the Transport Stream or Program Stream as indicated by the OCR_ES_ID field in the ES descriptor for that stream.

The following constraints shall apply for encoding of OCRs and FCRs in SL-packetized streams and FlexMux streams carried over an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream :

-    The OCRs and FCRs in each SL-packetized stream and each FlexMux stream associated to the same scene shall have the same resolution.
-    The resolution of OCRs and FCRs for a scene, f_{cr}, shall have a value smaller than or equal to 90 000 Hz.
-    The ratio $(f_{stc}(t) / 300) / f_{cr}$, shall be an integer value larger than or equal to one. Consequently the resolution of the OCR and FCR syntax elements may only take values such as 90 000 Hz, 45 000 Hz, 30 000 Hz, 22 500 Hz, 18 000 Hz, etc.

Within the above constraints and the ISO/IEC 14496-1 constraint that the resolution f_{cr} shall represent an integer number of cycles per second, f_{cr} can be selected as appropriate for the scene.

The ISO/IEC 14496 time stamps coded in the SL packet header shall refer to instants of the object time base of the stream carried in the SL packet. The resolution of each such time stamp shall be of a factor 2^k smaller than the resolution of the OCRs or FCRs associated to the stream, with k a positive integer larger than or equal to zero. To achieve the same wrap around, the length of the time stamp fields, TimeStampLength, shall be k bit smaller than the length of the OCR or FCR field, OCRLength and FCRLength, respectively. Hence for each stream the following conditions shall apply for encoding of time stamps :

-    TimeStampResolution = (OCRResolution or FCRResolution respectively) / 2^k, with k a positive integer larger than or equal to zero. ISO/IEC 14496-1 requires TimeStampResolution to represent an integer number of cycles per second.
-    TimeStampLength = OCRLength or FCRLength respectively - k.

The relationship between a value of the STC and the corresponding value of the object time base of a stream is established by associating PTS fields in PES packet headers with the OCR or FCR in SL packet headers and FlexMux Stream packets, respectively, as specified in 2.11.3.6 and 2.11.3.7.

2.11.3.4   Delivery timing of SL-packetized streams

To carry ISO/IEC 14496 content in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream, ISO/IEC 14496-1 SL-packetized streams are used. In each SL-packetized stream carried in a PES packet without the use of FlexMux, the objectClockReference field shall be encoded as follows:

1. An objectClockReference (OCR) field shall be present in the first SL packet header of a SL-packetized stream.

2. The SL-packetized stream shall be constructed such that the time interval between the bytes containing the last bit of successive OCR fields shall be less than or equal to 0,7 second. Thus:

    $$| \, t(i'') \, - \, t(i') \, | \; < = \; 0{,}7 \; s$$

    for all i' and i" where i' and i" are the indexes of the bytes containing the last bit of consecutive OCR fields in the FlexMux stream.

If an objectClockReference is encoded in an SL packet header, also the instantBitrate field shall be coded.

2.11.3.5   Delivery timing of FlexMux streams

Next to SL-packetized streams also the ISO/IEC 14496-1 FlexMux tool may be used to carry ISO/IEC 14496 content in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 streams. The payload of FlexMux packets shall consist of SL packets as specified in ISO/IEC 14496-1. In each FlexMux stream carried in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 stream the fmxClockReference field shall be encoded as follows :

1. An fmxClockReference (FCR) field shall be present in the first FlexMux packet of a FlexMux stream.

2. The FlexMux stream shall be constructed such that the time interval between the bytes containing the last bit of successive FCR fields shall be less than or equal to 0,7 second. Thus :

    $$| \, t(i'') \, - \, t(i') \, | \; < = \; 0{,}7 \; s$$

    for all i' and i" where i' and i" are the indexes of the bytes containing the last bit of consecutive FCR fields in the FlexMux stream.

3. All ISO/IEC 14496 time stamps within the SL-packetized streams carried within a FlexMux stream shall refer to instants of the object time base conveyed by the FCR fields in the FlexMux stream. The SL-packetized streams carried in FlexMux packets need not carry OCR fields. If OCR fields are present, they may be ignored.

2.11.3.6   Carriage of SL-packetized streams in PES packets

A single ISO/IEC 14496-1 SL-packetized stream may be mapped into a single PES stream. One and only one SL packet from an SL-packetized stream shall constitute the payload of one PES packet. PES packets that carry an SL-packetized stream shall be identified by stream_id = 0xFA in the PES packet header.

When an OCR field is coded in the SL packet header, a PTS shall be encoded in the header of the PES packet that carries such SL packet header. This PTS shall be encoded with the 33 bit value of the 90 kHz portion of the STC that corresponds to the value of the object time base at the instant in time indicated by the OCR.

The ES_ID associated to the SL-packetized stream shall be signaled by an SL descriptor as specified in 2.6.46.

2.11.3.7   Carriage of FlexMux streams in PES packets

PES packets with a payload consisting of FlexMux packets shall be identified by stream_id = 0xFB in the PES packet header. An integer number of FlexMux packets shall constitute the payload of one PES packet, i.e., the payload of a PES packet carrying a FlexMux stream shall start with a FlexMux packet header and shall end with the last byte of a FlexMux packet.

If an fmxClockReference (FCR) field is encoded in one of the FlexMux packets contained in a PES packet, then a PTS shall be encoded in the header of the PES packet that contains such FlexMux packet. This PTS shall be encoded with the 33 bit value of the 90 kHz portion of the STC that corresponds to the value of the object time base of the FlexMux stream at the instant in time indicated by the FCR. In case multiple FlexMux packets with an encoded FCR field are contained in a PES packet, the PTS shall correspond to the time indicated by the FCR in the first such FlexMux packet encountered in the payload of the PES packet.

The ES_IDs associated to each SL-packetized stream conveyed in the FlexMux stream shall be signaled by an FMC descriptor as specified in 2.6.38.

2.11.3.8   Carriage of SL packets and FlexMux packets in sections

For transport of ISO/IEC 14496 content in sections, ISO_IEC_14496_sections are defined. Only SL-packetized object descriptor streams and scene description streams shall use ISO_IEC_14496_sections. A single ISO_IEC_14496_section shall contain either an entire SL packet of an SL-packetized stream or an integer number of FlexMux packets each carrying an SL packet of the same ISO/IEC 14496-1 elementary stream.

Table 2-63 shows the syntax of ISO_IEC_14496_sections defined to convey ISO/IEC 14496-1 elementary streams, qualified by the table_id as either object descriptor or scene description stream data. Object descriptor stream data consists of an Object Descriptor Table that comprises a number of object descriptors. The Object Descriptor Table may be transmitted in multiple ISO_IEC_14496_sections. Scene description data consists of a Scene Description Table that may comprise a number of BIFS commands. The Scene Description Table may be transmitted in multiple ISO_IEC_14496_sections. It is not required that a complete table be received in order to process its payload. However, the payload of sections shall be processed in the correct order, as indicated by the value of the section_number field in the ISO_IEC_14496_section header bytes.

ISO/IEC 13818-1:1996/Amd.7:199x(E)

Table 2-63 — Section syntax for transport of ISO/IEC14496 streams

Syntax	No. of bits	Mnemonic
ISO_IEC_14496_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
ISO_IEC_14496_section_length	12	uimsbf
table_id_extension	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
if (PMT_has_SL_descriptor(current_PID)) {		
SL_Packet()		
}		
else if (PMT_has_FMC_descriptor(current_PID)) {		
for (i=1; i<N; i++)		
FlexMuxPacket()		
}		
else {		
for (i=1; i<N; i++)		
reserved	8	bslbf
}		
CRC_32	32	rpchof
}		

18

ITU-T Rec. H.222.0 (1996)/Amd.7(199x E)

table_id – This 8-bit field shall be set to '0x04' or '0x05' in case of an ISO_IEC_14496_section. A value of '0x04' indicates an ISO_IEC_14496_scene_description_section that carries an ISO/IEC 14496-1 scene description stream. A value of '0x05' indicates an ISO_IEC_14496_object_descriptor_section that carries an ISO/IEC 14496-1 object descriptor stream.

section_syntax_indicator – This 1-bit field shall be set to '1'.

private_indicator - This 1-bit field shall not be specified by this Specification

ISO_IEC_14496_section_length - This 12-bit field shall specify the number of remaining bytes in the section immediately following the ISO_IEC_14496_section_length field up to the end of the ISO_IEC_14496_section. The value of this field shall not exceed 4093 (0xFFD)

table_id_extension - This 16-bit field shall not be specified by this Specification; its use and value are define by the user.

version_number – This 5-bit field shall represent the version number of the Object Descriptor Table or Scene Description Table respectively. The version number shall be incremented by 1 modulo 32 with each new version of the table. Version control is at the discretion of the application.

current_next_indicator - This 1-bit field shall be set to 1.

section_number - This 8-bit field shall represent the number of the ISO_IEC_14496_section. The section_number field of the first ISO_IEC_14496_section of the Object Descriptor Table or the Scene Description Table shall have a value equal to 0x00. The value of section_number shall be incremented by 1 with each additional section in the table.

last_section_number - This 8-bit field shall specify the number of the last section of the Object Descriptor Table or Scene Description Table of which this section is a part.

PMT_has_SL_descriptor(current_PID) – a pseudo function that shall be true if an SL descriptor is contained in the descriptor loop in the Program Map Table for the ISO/IEC 13818-1 program element that conveys this ISO_IEC_14496_section.

SL_Packet() – a sync layer packet as specified in subclause 10.2.2 of ISO/IEC 14496-1.

PMT_has_FMC_descriptor(current_PID) – a pseudo function that shall be true if an FMC descriptor is contained in the descriptor loop in the Program Map Table for the ISO/IEC 13818-1 program element that conveys this ISO_IEC_14496_section.

FlexMuxPacket() – a FlexMux packet as specified in subclause 11.2.4 of ISO/IEC 14496-1.

CRC_32 - This 32-bit field shall contain the CRC value that gives a zero output of the registers in the decoder defined in Annex A of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 after processing the entire ISO_IEC_14496_section.

2.11.3.9   T-STD extensions

2.11.3.9.1   *T-STD Model for 14496 content*

Figure 2-5 shows extensions of the Transport System Target Decoder for delivery of ISO/IEC 14496 program elements encapsulated in ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Streams.
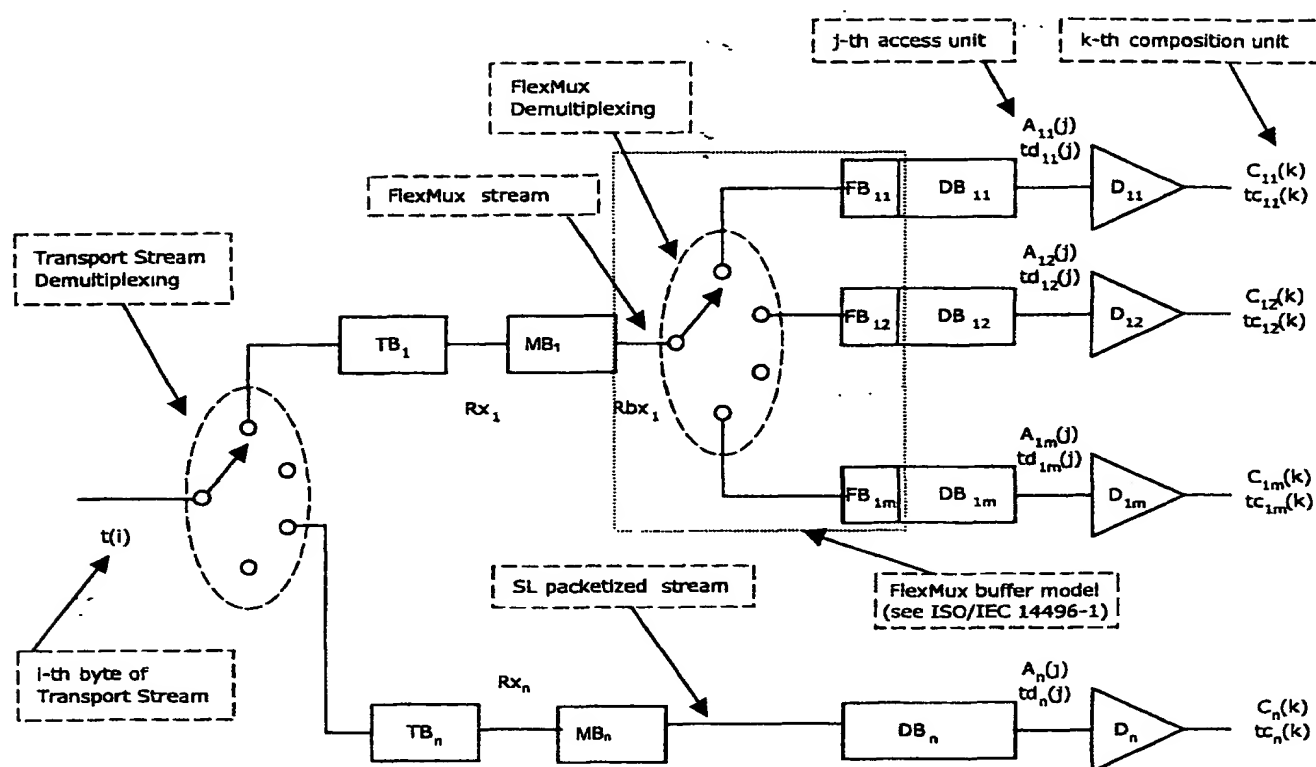
ISO/IEC 13818-1:1996/Amd.7:199x(E)

Figure 2-5 – T-STD model for ISO/IEC 14496 content

The following notation is used in Figure 2-5 and its description :

TB_n　　is the transport buffer.

MB_n　　is the multiplex buffer for FlexMux stream n or for SL-packetized stream n.

FB_{np}　is the FlexMux buffer for the elementary stream in FlexMux channel p of FlexMux stream n

DB_{np}　Is the decoder buffer for the elementary stream in FlexMux channel p of FlexMux stream n

DB_n　　is the decoder buffer for elementary stream n.

D_{np}　is the decoder for the elementary stream in FlexMux channel p of FlexMux stream n

D_n　　is the decoder for elementary stream n.

Rx_n　is the rate at which data are removed from TB_n.

Rbx_n　is the rate at which data are removed from MB_n.

$A_{np}(j)$　is the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n. $A_{np}(j)$ is indexed in decoding order.

$A_n(j)$　is the jth access unit in elementary stream n. $A_n(j)$ is indexed in decoding order.

$Td_{np}(j)$　is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n.

$Td_n(j)$　is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream n.

$C_{np}(k)$　is the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n. $C_{np}(k)$ results from decoding $A_{np}(j)$. $C_{np}(k)$ is indexed in composition order.

$C_n(k)$　is the kth composition unit in elementary stream n. $C_n(k)$ results from decoding $A_n(j)$. $C_n(k)$ is indexed in composition order.

$tc_{np}(k)$　is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n.

tc$_n$(k) is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream n.

t(i) Indicates the time in seconds at which the ith byte of the Transport Stream enters the system target decoder.

2.11.3.9.2 Processing of FlexMux streams

Complete Transport Stream packets containing data from FlexMux stream n are passed to the transport buffer for FlexMux stream n, TB$_n$. The size of TB$_n$ is fixed at 512 bytes. All bytes that enter TB$_n$ are removed from TB$_n$ at a rate Rx$_n$, specified by the TB_leak_rate field in the MultiplexBuffer descriptor associated with FlexMux stream n. When there is no data in buffer TB$_n$, rate Rx$_n$ is equal to zero. Duplicate Transport Stream packets are not delivered to MB$_n$.

In case of carriage in PES packets, the PES packet header and payload data bytes are delivered to buffer MB$_n$; all other bytes leaving TB$_n$ do not enter MB$_n$, and may be used to control the system. In case of carriage in ISO_IEC_14496_sections, the section header, payload and CRC32 data bytes are delivered to buffer MB$_n$; all other bytes do not enter MB$_n$ and may be used to control the system. In either case, the size of MB$_n$ shall be specified by the MB_buffer_size field in the MultiplexBuffer descriptor.

The FlexMux Stream packet bytes in buffer MB$_n$ are all delivered to their associated FlexMux buffer at the rate specified by the field fmxRate encoded in the FlexMux stream and in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only FlexMux packet payload data bytes in FlexMux channel p of FlexMux stream n enter buffer FB$_{np}$. FlexMux packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system. The rate specified by the fmxRate field shall be applicable for all FlexMux packets in the stream immediately following the FlexMux Clock Reference channel packet up to the next encountered FlexMux Clock Reference packet. When there is no FlexMux stream data present in MB$_n$, no data is removed from MB$_n$. Bytes from the PES packet header or from the ISO_IEC_14496_section header that immediately precede a FlexMux header are instantaneously removed and discarded and may be used to control the system. Bytes from the ISO_IEC_14496_section CRC32 fields that immediately follow the last FlexMux Stream packet in the section payload are removed instantaneously and discarded and may be used to verify the integrity of the data. Bytes from the FlexMux Clock Reference channel are instantaneously removed and discarded and may be used to lock the ISO/IEC 14496 object time base to the STC. When there is no PES packet or section payload data bytes, respectively present in MB$_n$, no data is removed from MB$_n$. All data that enters MB$_n$ leaves it. All PES packet payload bytes of stream n enter the FlexMux demultiplexer instantaneously upon leaving MB$_n$.

2.11.3.9.3 Definition of FlexMux Buffer, FB$_{np}$

For each channel p of a FlexMux stream n, the size of FlexMux buffer FB$_{np}$ is defined using the FmxBufferSize descriptor. FlexMux packet payload bytes are transferred from buffer FB$_{np}$ to decoder buffer DB$_{np}$ in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only SL packet payload bytes in FlexMux channel p of FlexMux stream n enter buffer DB$_{np}$. The SL packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system.

2.11.3.9.4 Processing of SL-packetized streams

Complete Transport Stream packets containing data from SL-packetized stream n are passed to the transport buffer for SL-packetized stream n, TB$_n$. All bytes that enter TB$_n$ are removed at a rate Rx$_n$, specified by the TB_leak_rate field in the MultiplexBuffer descriptor. When there is no data in buffer TB$_n$, rate Rx$_n$ is equal to zero. Duplicate Transport Stream packets are not delivered to MB$_n$.

In case of carriage in PES packets, the PES packet header and payload data bytes are delivered to buffer MB$_n$; all other bytes leaving TB$_n$ do not enter MB$_n$, and may be used to control the system. In case of carriage in ISO_IEC_14496_sections, the section header, payload and CRC32 data bytes are delivered to
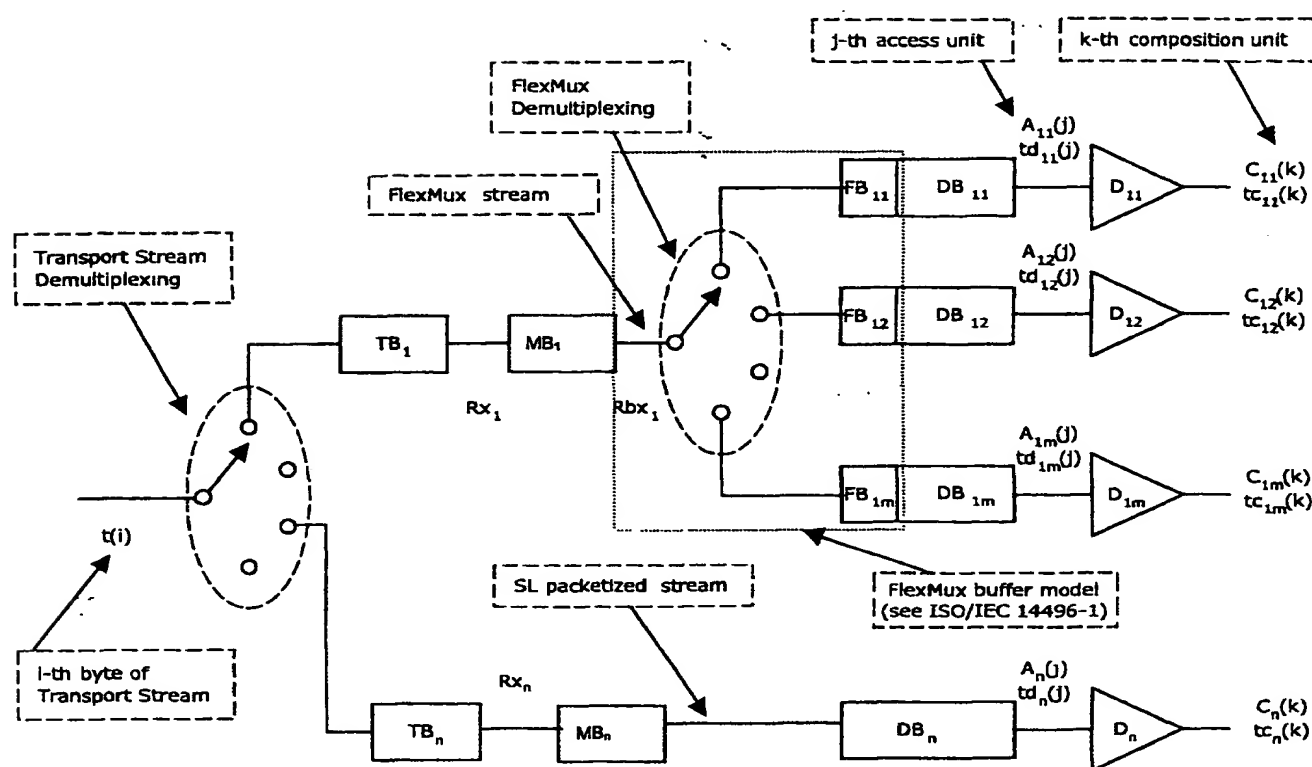
Figure 2-5 - T-STD model for ISO/IEC 14496 content

The following notation is used in Figure 2-5 and its description :

TB_n	is the transport buffer.
MB_n	is the multiplex buffer for FlexMux stream n or for SL-packetized stream n.
FB_{np}	is the FlexMux buffer for the elementary stream in FlexMux channel p of FlexMux stream n
DB_{np}	Is the decoder buffer for the elementary stream in FlexMux channel p of FlexMux stream n
DB_n	is the decoder buffer for elementary stream n.
D_{np}	is the decoder for the elementary stream in FlexMux channel p of FlexMux stream n
D_n	is the decoder for elementary stream n.
Rx_n	is the rate at which data are removed from TB_n.
Rbx_n	is the rate at which data are removed from MB_n.
$A_{np}(j)$	is the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n. $A_{np}(j)$ is indexed in decoding order.
$A_n(j)$	is the jth access unit in elementary stream n. $A_n(j)$ is indexed in decoding order.
$Td_{np}(j)$	is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n.
$Td_n(j)$	is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream n.
$C_{np}(k)$	is the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n. $C_{np}(k)$ results from decoding $A_{np}(j)$. $C_{np}(k)$ is indexed in composition order.
$C_n(k)$	is the kth composition unit in elementary stream n. $C_n(k)$ results from decoding $A_n(j)$. $C_n(k)$ is indexed in composition order.
$tc_{np}(k)$	is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n.

$tc_n(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream n.

$t(i)$ Indicates the time in seconds at which the ith byte of the Transport Stream enters the system target decoder.

2.11.3.9.2   Processing of FlexMux streams

Complete Transport Stream packets containing data from FlexMux stream n are passed to the transport buffer for FlexMux stream n, TB_n. The size of TB_n is fixed at 512 bytes. All bytes that enter TB_n are removed from TB_n at a rate Rx_n, specified by the TB_leak_rate field in the MultiplexBuffer descriptor associated with FlexMux stream n. When there is no data in buffer TB_n, rate Rx_n is equal to zero. Duplicate Transport Stream packets are not delivered to MB_n.

In case of carriage in PES packets, the PES packet header and payload data bytes are delivered to buffer MB_n; all other bytes leaving TB_n do not enter MB_n, and may be used to control the system. In case of carriage in ISO_IEC_14496_sections, the section header, payload and CRC32 data bytes are delivered to buffer MB_n; all other bytes do not enter MB_n and may be used to control the system. In either case, the size of MB_n shall be specified by the MB_buffer_size field in the MultiplexBuffer descriptor.

The FlexMux Stream packet bytes in buffer MB_n are all delivered to their associated FlexMux buffer at the rate specified by the field fmxRate encoded in the FlexMux stream and in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only FlexMux packet payload data bytes in FlexMux channel p of FlexMux stream n enter buffer FB_{np}. FlexMux packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system. The rate specified by the fmxRate field shall be applicable for all FlexMux packets in the stream immediately following the FlexMux Clock Reference channel packet up to the next encountered FlexMux Clock Reference channel packet. When there is no FlexMux stream data present in MB_n, no data is removed from MB_n. Bytes from the PES packet header or from the ISO_IEC_14496_section header that immediately precede a FlexMux header are instantaneously removed and discarded and may be used to control the system. Bytes from the ISO_IEC_14496_section CRC32 fields that immediately follow the last FlexMux Stream packet in the section payload are removed instantaneously and discarded and may be used to verify the integrity of the data. Bytes from the FlexMux Clock Reference channel are instantaneously removed and discarded and may be used to lock the ISO/IEC 14496 object time base to the STC. When there is no PES packet or section payload data bytes, respectively present in MB_n, no data is removed from MB_n. All data that enters MB_n leaves it. All PES packet payload bytes of stream n enter the FlexMux demultiplexer instantaneously upon leaving MB_n.

2.11.3.9.3   Definition of FlexMux Buffer, FB_{np}

For each channel p of a FlexMux stream n, the size of FlexMux buffer FB_{np} is defined using the FmxBufferSize descriptor. FlexMux packet payload bytes are transferred from buffer FB_{np} to decoder buffer DB_{np} in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only SL packet payload bytes in FlexMux channel p of FlexMux stream n enter buffer DB_{np}. The SL packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system.

2.11.3.9.4   Processing of SL-packetized streams

Complete Transport Stream packets containing data from SL-packetized stream n are passed to the transport buffer for SL-packetized stream n, TB_n. All bytes that enter TB_n are removed at a rate Rx_n, specified by the TB_leak_rate field in the MultiplexBuffer descriptor. When there is no data in buffer TB_n, rate Rx_n is equal to zero. Duplicate Transport Stream packets are not delivered to MB_n.

In case of carriage in PES packets, the PES packet header and payload data bytes are delivered to buffer MB_n; all other bytes leaving TB_n do not enter MB_n, and may be used to control the system. In case of carriage in ISO_IEC_14496_sections, the section header, payload and CRC32 data bytes are delivered to

buffer MB_n; all other bytes do not enter MB_n and may be used to control the system. In either case the size of MB_n is specified by the MB_buffer_size field in the MultiplexBuffer descriptor.

The SL-packetized stream bytes in buffer MB_n are all delivered to the decoder buffer DBn at the rate specified by the field instantBitRate encoded in the SL-packetized stream and in compliance with the System Decoder Model defined in clause 7.4 of ISO/IEC 14496-1. The rate specified by the instantBitRate field shall be applicable for all data bytes in the SL-packetized stream immediately following the instantBitRate field in the SL packet header up to the next encountered instantBitRate field. If there are no SL-packetized stream bytes in MB_n, no bytes are removed from MB_n. Bytes from the PES packet header or from the ISO_IEC_14496_section header that immediately precede a SL packet header are instantaneously removed and discarded and may be used to control the system. Bytes from the ISO_IEC_14496_section CRC32 fields that immediately follow the last SL packet payload byte in the section are removed instantaneously and discarded and may be used to verify the integrity of the data. When there are no PES packet or section payload data bytes, respectively present in MB_n, no data is removed from MB_n. All data that enters MB_n leaves it. All PES packet payload bytes of stream n enter buffer DB_n instantaneously upon leaving MB_n, with the exception of the SL packet headers. Bytes from the SL packet headers do not enter DB_n and may be used to control the system. The size of decoder buffer DB_n is given by the bufferSizeDB of the DecoderConfigDescriptor defined in ISO/IEC 14496-1.

2.11.3.9.5   Buffer management

Transport streams shall be constructed so that conditions defined in this subclause are satisfied.

TB_n shall not overflow and shall be empty at least once every second. MB_n shall not overflow. FB_{np} shall not overflow. DB_{np} and DB_n shall neither underflow nor overflow. Underflow of DB_{np} occurs when one or more bytes of an access unit are not present in DB_{np} at the decoding time associated with this access unit. Underflow of DB_n occurs when one or more bytes of an access unit are not present in DB_n at the decoding time associated with this access unit.

2.11.3.10   Carriage within a Transport Stream

2.11.3.10.1   Overview
A Transport Stream may contain one or more programs, each described by a Program Map Table. ISO/IEC 14496 content can be conveyed in addition to the already defined stream types for such a program. Elements of the ISO/IEC 14496 content may be conveyed in one or more ISO/IEC 13818-1 program elements referenced by a unique PID value within a Transport Stream. As a special case, it is possible that a program within a Transport Stream consists only of ISO/IEC 14496 program elements. ISO/IEC 14496 content associated to a program and carried in the Transport Stream shall be referenced in the Program Map Table of that program. An initial object descriptor shall be used to define an ISO/IEC 14496-1 scene; the use of this descriptor is specified in clause 2.11.3.10.2.

Carriage of ISO/IEC 14496 content in a PID is signaled by a stream_type value of 0x12 or 0x13 in the Program Map Table in association with that PID value. A value of 0x12 indicates carriage in PES packets. The stream_id field in the PES packet header signals whether the PES packet contains a single SL packet or a number of FlexMux packets. A stream_type value of 0x13 in the Program Map Table indicates that the program element carries an object descriptor stream or a BIFS-Command stream contained in sections. In this case the table_id in the section header indicates whether an object descriptor stream is carried in the sections or a BIFS-Command stream. See also table 2-64. The section contains either a single SL packet or a number of FlexMux packets, as indicated by the presence of an SL descriptor or a FMC descriptor respectively in the descriptor loop of the Program Map Table for the ISO/IEC 13818-1 program element that carries the sections. When ISO/IEC 14496 content is carried, the SL descriptor and the FMC descriptor shall specify the ES_ID for each encapsulated ISO/IEC 14496 stream. When the assignment of ES_ID values changes, the Program Map Table shall be updated and the version_number of the PMT shall be incremented by 1 modulo 32. An example of a content access procedure for ISO/IEC 14496 program components within a Transport Stream is given in Annex R Carriage of ISO/IEC 14496 scenes in ITU-T Rec. H.222.0 | ISO/IEC 13818-1.

Table 2–64 – ISO/IEC defined options for carriage of an ISO/IEC 14496 scene and associated streams in ITU-T Rec. H.222.0 | ISO/IEC 13818-1

ISO/IEC 14496-1 object descriptor streams	Encapsulation in SL packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1010'
		Carriage in ISO_IEC_14496_sections	Stream_type = 0x13	Table_id = 0x05
	Encapsulation in SL packets followed by Multiplex into FlexMux packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1011'
		Carriage in ISO_IEC_14496_sections	Stream_type = 0x13	Table_id = 0x05
ISO/IEC 14496-1 scene description streams	Encapsulation in SL packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1010'
		Carriage in ISO_IEC_14496_sections	Stream_type = 0x13	Table_id = 0x04
	Encapsulation in SL packets followed by Multiplex into FlexMux packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1011'
		Carriage in ISO_IEC_14496_sections	Stream_type = 0x13	Table_id = 0x04
All other ISO/IEC 14496 streams	Encapsulation in SL packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1010'
	Encapsulation in SL packets followed by Multiplex into FlexMux packets	Carriage in PES packets	Stream_type = 0x12	Stream_id = '1111 1011'

2.11.3.10.2   Initial Object Descriptor

In case of carriage of an ISO/IEC 14496-1 scene, the ISO/IEC 14496-1 initial object descriptor serves as the initial access point to all associated streams. The initial object descriptor shall be conveyed in the IOD descriptor located in the descriptor loop immediately following the program_info_length field in the Program Map Table of the program to which the scene is associated. It contains ES_Descriptors identifying the scene description and object descriptor streams that form part of this program. It may also contain ES_Descriptors identifying one or more associated IPMP or OCI streams. Identification of streams is done by means of ES_IDs as specified in clause 8 of ISO/IEC 14496-1.

2.11.3.11 P-STD Model for 14496 content

Figure 2-6 shows the STD model when ISO/IEC 14496 systems data are carried in a Program Stream.
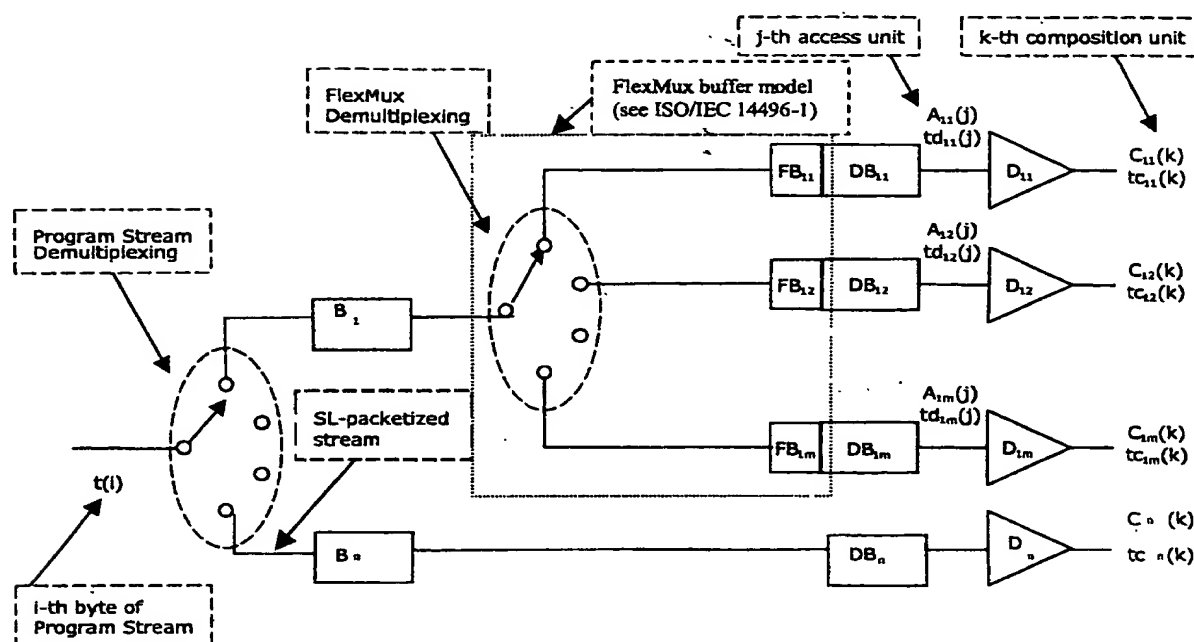
Figure 2-6 – P-STD model for ISO/IEC 14496 Systems stream

The following notation is used in Figure 2-6 and its description :

B_n is the input buffer for FlexMux stream n or for SL-packetized stream n.

FB_{np} is the FlexMux buffer for the elementary stream in FlexMux channel p of FlexMux stream n

DB_{np} is the decoder buffer for the elementary stream in FlexMux channel p of FlexMux stream n

DB_n is the decoder buffer for elementary stream n.

D_{np} is the decoder for elementary stream in FlexMux channel p of FlexMux stream n

D_n is the decoder for elementary stream n.

$A_{np}(j)$ is the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n. $A_{np}(j)$ is indexed in decoding order.

$A_n(j)$ is the jth access unit in elementary stream n. An(j) is indexed in decoding order.

$Td_{np}(j)$ is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream in FlexMux channel p of FlexMux stream n.

$Td_n(j)$ is the decoding time, measured in seconds, in the system target decoder of the jth access unit in elementary stream n.

$C_{np}(k)$ is the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n. $C_{np}(k)$ results from decoding $A_{np}(j)$. $C_{np}(k)$ is indexed in composition order.

$C_n(k)$ is the kth composition unit in elementary stream n. $C_n(k)$ results from decoding $A_n(j)$. $C_n(k)$ is indexed in composition order.

$tc_{np}(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream in FlexMux channel p of FlexMux stream n.

$tc_n(k)$ is the composition time, measured in seconds, in the system target decoder of the kth composition unit in elementary stream n.

$t(i)$ Indicates the time in seconds at which the ith byte of the Program Stream enters the system target decoder.

2.11.3.11.1 Processing of FlexMux streams

At the input of the STD each byte in the payload of PES packets carrying a FlexMux stream n is transferred instantaneously to buffer B_n. The i-th byte enters B_n at time $t(i)$. PES packet header bytes do not enter buffer B_n and may be used to control the system. The size of B_n is specified by the P-STD_buffer_size field in the header of the PES packet that carries stream n.

The FlexMux stream packet bytes in buffer B_n are all delivered to their associated FlexMux buffer at the rate specified by the field fmxRate encoded in the FlexMux stream and in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only FlexMux packet payload data bytes in FlexMux channel p of FlexMux stream n enter buffer FB_{np}. FlexMux packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system. The rate specified by the fmxRate field shall be applicable for all FlexMux packets in the stream up to the next encountered FlexMux Clock Reference channel packet. Bytes from the FlexMux Clock Reference channel are instantaneously removed and discarded and may be used to lock the ISO/IEC 14496 object time base to the STC. When there is no PES packet payload data present in B_n, no data is removed from B_n. All data that enters B_n leaves it. All PES packet payload bytes of stream n enter the FlexMux demultiplexer instantaneously upon leaving B_n.

2.11.3.11.2 Definition of FlexMux Buffer, FB_{np}

For each channel p of a FlexMux stream n, the size of FlexMux buffer FB_{np} is defined using the FmxBufferSize descriptor if a Program Stream Map is present in the Program Stream. FlexMux packet payload bytes are transferred from buffer FB_{np} to decoder buffer DB_{np} in compliance with the FlexMux buffer model defined in clause 11.2.9 of ISO/IEC 14496-1. Only SL packet payload bytes in FlexMux channel p of FlexMux stream n enter buffer DB_{np}. The SL packet header bytes in FlexMux channel p of FlexMux stream n are discarded and may be used to control the system

2.11.3.11.3 Processing of SL-packetized streams

At the input of the STD each byte in the payload of PES packets carrying an SL-packetized stream n is transferred instantaneously to buffer B_n. The i-th byte enters B_n at time $t(i)$. PES packet header bytes do not enter buffer B_n and may be used to control the system. The size of B_n is specified by the P-STD_buffer_size field in the header of the PES packet that carries stream n. The SL-packetized stream bytes in buffer B_n are delivered to the decoder buffer DBn at the rate specified by the field instantBitRate encoded in the SL-packetized stream and in compliance with the System Decoder Model defined in clause 7.4 of ISO/IEC 14496-1. The rate specified by the instantBitRate field shall be applicable for all data bytes in the SL-packetized stream up to the next encountered instantBitRate field. When there is no PES packet payload data present in B_n, no data is removed from B_n. All data that enters B_n leaves it. All bytes of stream n enter buffer DB_n instantaneously upon leaving B_n, with the exception of the SL packet headers. Bytes from the SL packet headers do not enter DB_n and may be used to control the system. The size of decoder buffer DB_n is given by the bufferSizeDB of the DecoderConfigDescriptor defined in ISO/IEC 14496-1.

2.11.3.11.4 Buffer management

Program Streams shall be constructed so that B_n does not overflow. FB_{np} shall not overflow. DB_{np} and DB_n shall neither underflow nor overflow. Underflow of DB_{np} occurs when one or more bytes of an access unit are not present in DB_{np} at the decoding time associated with this access unit. Underflow of DB_n occurs when one or more bytes of an access unit are not present in DB_n at the decoding time associated with this access unit

2.11.3.12　Carriage within a Program Stream

2.11.3.12.1　Overview

A Program Stream contains only one program. ISO/IEC 14496 data can be conveyed in addition to the already defined stream types for such a program. As a special case, it is also possible that a Program Stream carries only ISO/IEC 14496 data. If a Program Stream Map is present, ISO/IEC 14496 content carried in the Program Stream shall be referenced as follows. Carriage of ISO/IEC 14496-1 scenes and associated ISO/IEC 14496 streams in SL and FlexMux packets is indicated by the appropriate stream_id and by an initial object descriptor; the use of this descriptor is specified in clause 2.11.3.12.2. For each carried ISO/IEC 14496 stream the SL descriptor and the FMC descriptor shall specify the ES_ID. When the assignment of ES_ID values changes, the Program Stream Map, if present, shall be updated and the program_stream_map_version shall be incremented by 1 modulo 32. Note that in a Program Stream the ISO/IEC 14496 content may also be referenced by private means.

For an example of a content access procedure for ISO/IEC 14496 program components within a Program Stream, see Annex R.

2.11.3.12.2　Initial Object Descriptor

In case of carriage of an ISO/IEC 14496-1 scene, the ISO/IEC 14496 initial object descriptor serves as the initial access point to all associated streams. If a Program Stream Map is present in the Program Stream, the initial object descriptor shall be conveyed in the IOD descriptor that is located in the descriptor loop immediately following the program_stream_info_length field. It contains ES_Descriptors identifying the scene description and object descriptor streams of the scene that form part of this program. It may also contain ES_Descriptors identifying one or more associated IPMP or OCI streams. Identification of streams is done by means of ES_IDs as specified in clause 8 of ISO/IEC 14496-1. In a Program Stream, the initial object descriptor may also be conveyed by private means.

EP00400840.5

9) Add after Annex Q:

"

Annex R Carriage of ISO/IEC 14496 scenes in ITU-T Rec. H.222.0 | ISO/IEC 13818-1

(This annex does not form an integral part of this Recommendation | International Standard)

R.1 Content access procedure for ISO/IEC 14496 program components within a Program Stream

The following provides a reference receiver acquisition procedure for accessing ISO/IEC 14496 program elements in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Program Stream. Here, it is assumed that the Program Stream includes a Program Stream Map (conveyed in a PES packet having stream_id equal to 0xBC):

1. Acquire the Program Stream Map
2. Identify the IOD descriptor in the first descriptor loop
3. Identify the ES_IDs of the object descriptor, scene description and other streams described within the initial object descriptor
4. Acquire the SL descriptor and FMC descriptor in the second descriptor loop for elementary_stream_ids 0xFA and 0xFB, as applicable.
5. Generate from these descriptors a stream map table between ES_IDs and associated elementary_stream_id plus FlexMux channel, if applicable.
6. Locate the object descriptor stream using its ES_ID and the stream map table.
7. Locate other streams described in the initial object descriptor using their ES_ID and the stream map table.
8. Continuously monitor the object descriptor stream and identify ES_IDs of additional streams.
9. Locate the additional streams using their ES_ID and the stream map table.

Figure R-1 gives an example of ISO/IEC 14496 content in a Program Stream, consisting of object descriptor stream, scene description stream (BIFS-Command), BIFS-Anim stream and IPMP stream. All ISO/IEC 14496 streams are multiplexed in a single FlexMux stream.

ISO/IEC 13818-1:1996/Amd.7:199x(E)
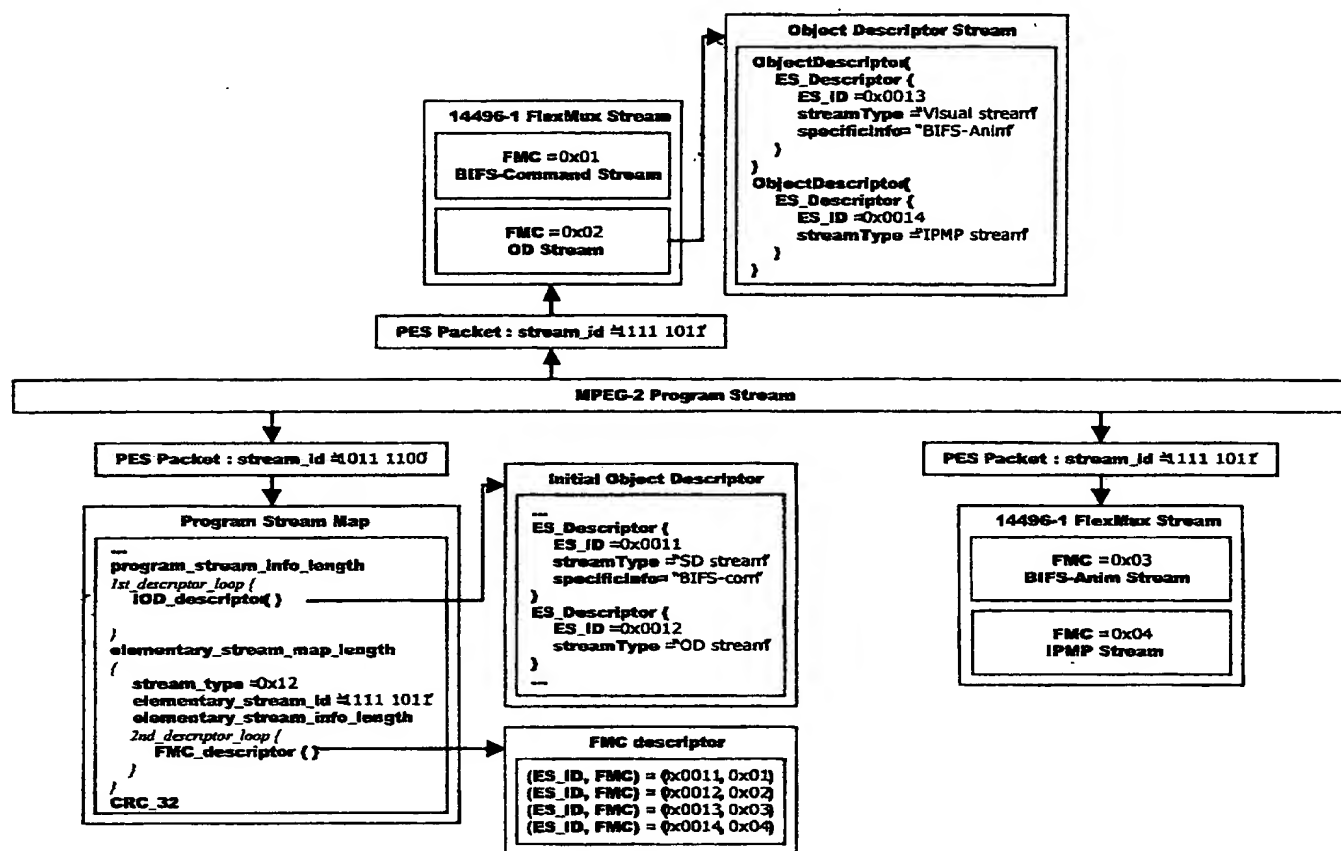
Figure R-1 - Example of ISO/IEC 14496 content in a Program Stream

EP00400840.5

R.2 Content access procedure for ISO/IEC 14496 program components within a Transport Stream

The following provides a reference receiver acquisition procedure for accessing ISO/IEC 14496 program elements in an ITU-T Rec. H.222.0 | ISO/IEC 13818-1 Transport Stream:

1. Acquire the Program Map Table for the desired program
2. Identify the IOD descriptor in the first descriptor loop
3. Identify the ES_IDs of the object descriptor, scene description and other streams described within the initial object descriptor
4. Acquire the set of all SL descriptors and FMC descriptors present in the second descriptor loop for any of the elementary_PIDs.
5. Generate from these descriptors a stream map table between ES_IDs and associated elementary_PID plus FlexMux channel, if applicable.
6. Locate the object descriptor stream using its ES_ID and the stream map table.
7. Locate other streams described in the initial object descriptor using their ES_ID and the stream map table.
8. Continuously monitor the object descriptor stream and identify ES_IDs of additional streams.
9. Locate the additional streams using their ES_ID and the stream map table.

Figure R-2 gives an example of ISO/IEC 14496 program elements in a Transport Stream, consisting of object descriptor stream, scene description stream (BIFS-Command), BIFS-Anim and IPMP elementary streams. BIFS-Command and OD stream are conveyed by means of ISO_IEC_14496_sections, while BIFS-Anim and IPMP elementary streams are conveyed in PES packets referenced by two distinct elementary_PID values, without the use of the ISO/IEC 14496-1 FlexMux tool.
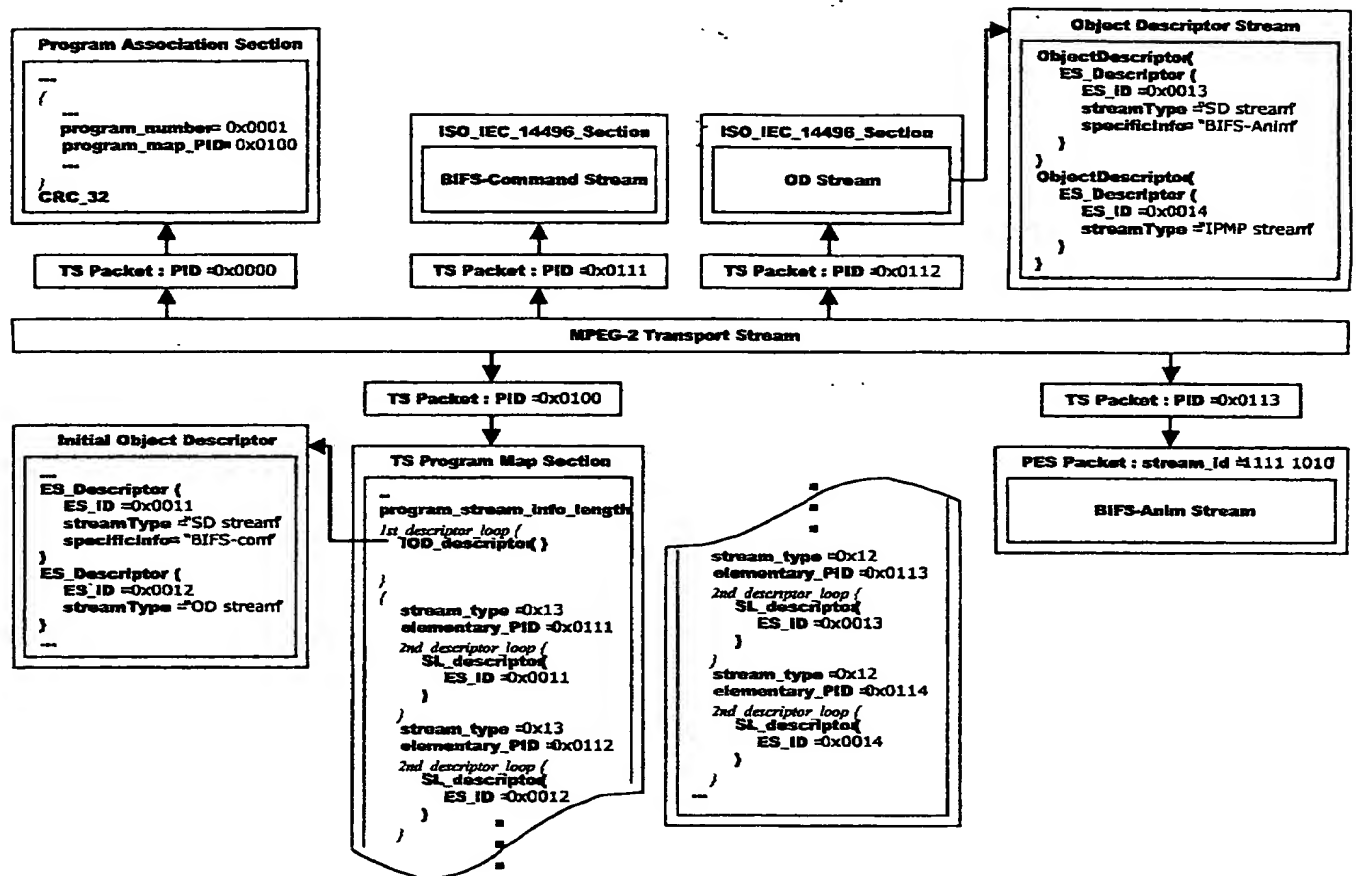
Figure R-2 - Example of ISO/IEC 14496 content in a Transport Stream

32

MPEG-4 ON MPEG-2 SYSTEM

ARCHITECTURE DOCUMENT

Authors : Laurent Herrmann, François Martin

Date : 15/03/2000

Project :	[96397]	Class :	602	N° :	[N° Doc]	Version :	[N° Ver]

Page 1 / 17

32

27-03-2000   LABORATOIRES   EP00400840.5 ONIQUE PHILIPS   DESC
22 AVENUE DESCARTES - BP 15 - F94453 LIMEIL   EL : 33 (0)1 45 10 67 00 - FAX · 33 (0)1 45 10 69 58

33

Table of contents

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

1. Introduction

1.1. Purpose and scope

1.2. Terminology

1.2.1. Definitions

1.2.2. Abbreviations and acronyms

BIFS	Binary Format for Scene description
EPG	Electronic Program Guide
MPEG	Moving Picture Expert Group
OD	Object Descriptor
PCR	Program Clock Reference
PES	Packet Elementary Stream
PSI	Program System Information

1.3. Reference documents

1.3.1. Controlling documents

[1] "*MPEG-4 Systems : Information Technology – Coding of Audiovisual Objects, Systems – ISO/IEC 14496-1*", May 1998

[2] "*MPEG-4 Video : Information Technology – Coding of Audiovisual Objects, Visual – ISO/IEC 14496-2*", May 1998

[3] "*MPEG-4 Audio : Information Technology – Coding of Audiovisual Objects, Audio – ISO/IEC 14496-3*", May 1998

[4] "*MPEG-2 Systems : Information Technology – Coding of Audiovisual Objects, Systems – ISO/IEC 13818-1*", November 1994

[5] "*MPEG-2 Systems Amendment 7 : Information Technology – Coding of Audiovisual Objects, Transport of MPEG-4 data over MPEG-2 Systems*", N3050, January 2000

1.3.2. Controlled documents

1.3.3. Applicable documents

1.4. Responsibility and approvals

1.4.1. Responsibility

The project leader is responsible for this document, the plans and the application of the plans.

1.4.2. Approvals

The group leader and the team involved in the development process are responsible for approvals

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

34

1.5. Overview

This document presents the architecture for the transport of MPEG-4 data in MPEG-2 Systems.

The end-to-end architecture is presented as well as the server and the terminal platforms.

Application scenarios and a proposed set-up demonstration are also described.

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

2. Transport of MPEG-4 over MPEG-2 End-to-End System Architecture Overview

2.1. Transport of MPEG-4 in MPEG-2 Systems Standard Specification

The standard specification is fully described in "*MPEG-2 Systems Amendment 7 : Information Technology – Coding of Audiovisual Objects, Transport of MPEG-4 data over MPEG-2 Systems*", N3050, January 2000. We propose here an architecture for validating this specification using an end-to-end system.

2.1.1. Transport of MPEG-4 Streams in MPEG-2 Systems

For "simple" MPEG-4 Audio and Video streams (without Scene description - BIFS - nor Object Descriptors), the method that is used is the same as for classical MPEG-2 A/V Streams. It consists in a mapping of MPEG-4 data in PES packets (MPEG-2 Systems Packetized Elementary Streams). A reference in the Program System Information (PSI table) is also needed to handle MPEG-4 stream type.

The use of MPEG-4 Systems Synchronization Layer (SL) and/or FlexMux (FM) is not mandatory.

2.1.2. Transport of MPEG-4 Scenes in MPEG-2 Systems

For more complex applications with MPEG-4 Scenes involving MPEG-4 A/V streams and BIFS and Object Descriptors, the streaming data (MPEG-4 A/V) is still mapped to PES packets but the use of SL/FM is mandatory for the systems encapsulation before the PES layer.

Moreover the BIFS and OD data are conveyed via MPEG-2 Sections (Tables that are re-transmitted regularly) using SL/FM format.

2.2. Global End-to-End System Architecture
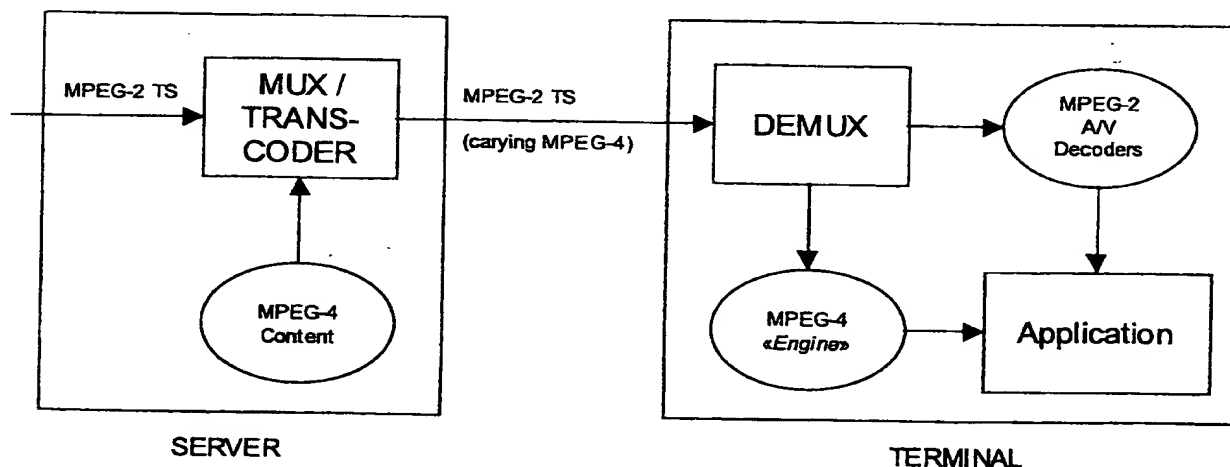
Figure 1 : Global End-to-End System Architecture for the transport of MPEG-4 data over MPEG-2 Systems

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

37

The end-to-end system is composed of 2 main parts : the Server and the Terminal. A network layer (MPEG-2 Transport Stream) is the link between these 2 parts.

The role of the Server is to generate an MPEG-2 Transport Stream composed of MPEG-2 and MPEG-4 A/V streams. The way of encapsulating MPEG-4 data in MPEG-2 Systems has to be compliant with the MPEG specification [5].

The role of the Terminal is to retrieve the MPEG-2 and MPEG-4 A/V data from an input MPEG-2 TS stream and to use these data in a client application.

2.3. Server Architecture

2.3.1. Platform Architecture

The Server Platform is described on Figure 2 below. It is based on a trans-coding platform.

The main input of this platform are :

- An existing MPEG-2 transport stream,
- MPEG-4 data (audio, video, BIFS, OD, ...).

The main output of the platform is a MPEG2- transport stream with MPEG-4 encapsulated in it.
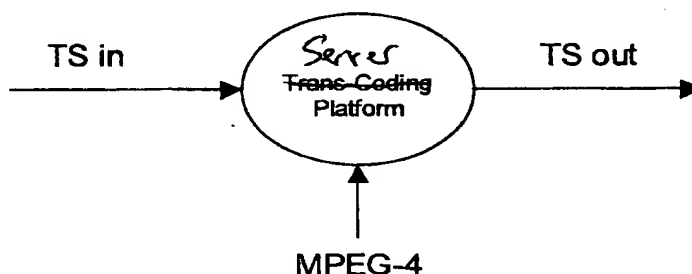
Figure 2 : Platform Architecture

The main blocks of such a system are described on Figure 3. These are :

- A bandwidth creation block that is in charge of creating bandwidth in the incoming transport stream in order to insert the MPEG-4 data,
- A MPEG-4 content creation block, that provide MPEG-4 content to be encapsulated,
- A MPEG-4 content encapsulation block that is in charge of the encapsulation of the MPEG-4 data, provided by the MPEG-4 content creation block, in the MPEG-2 transport stream, provided by the bandwidth creation block.

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

37

38
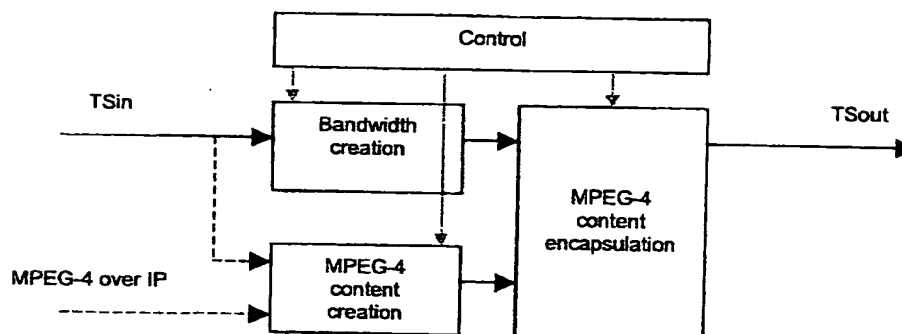
Figure 3: Details of the ~~Trans-Coding~~ *Server* Platform

There may be different ways to realize the functionality of the previous presented blocks. Some are detailed in the following sections.

2.3.2. Bandwidth Creation Block

Two ways may be considered to create bandwidth in the incoming stream. These are null transport packets insertion and bit rate trans-coding.

2.3.2.1. Null Packet Insertion

The method to create bandwidth consists in the insertion of null transport packet in the incoming stream. As a result the final output transport bit rate is higher than the incoming transport bit rate. An example of bit rate increment by insertion of null transport packet is shown on Figure 4. In that case the bit rate has been increase by a 3 factor.
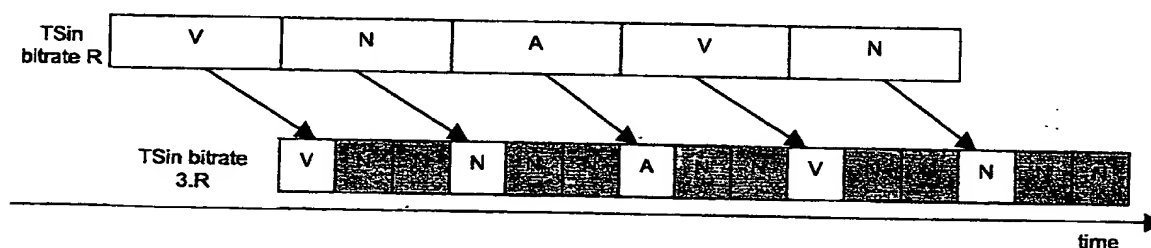
Figure 4 : Bandwidth Creation by the Insertion of Null TS Packet

If this method is used, due to packet contraction, care must be taken to correctly re-stamped PCR fields in transport packets carrying the PCR. This effect is shown on Figure 5.
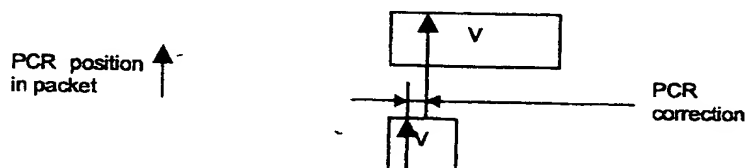
Figure 5 : PCR Correction

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

Page 9 / 17

38

39

2.3.2.2.   Bit-rate Trans-coding

When the output transport bit rate cannot be increased, the previous method cannot be used. In that case it is possible to provide bandwidth for the video data by the trans-coding one of the MPEG-2 video data.

Such a trans-coder is described on figure 6. In this system one of the program contains in the incoming transport stream is demux. Then its video is trans-coded in order to reduce its bit rate. Finally the video is remux in the incoming stream before going to the content encapsulation block.
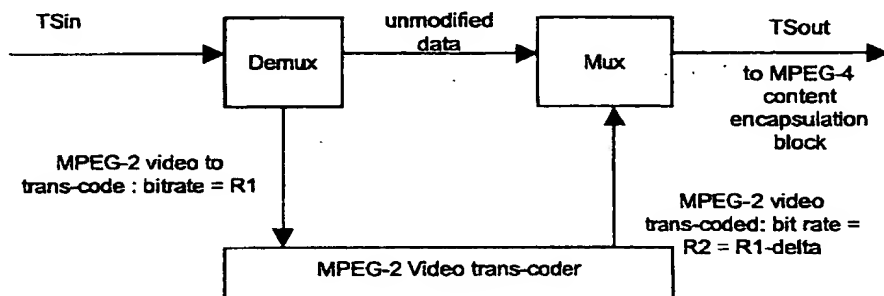
Figure 6: Use of MPEG-2 Video Trans-Coding for Bandwidth Creation

It should be noted that the mux part of the trans-coder might be a part of the MPEG-4 encapsulation block.

2.3.3.   MPEG-4 Content Creation Block

Different way (Figure 3) may be used to provide MPEG-4 content (related to the MPEG-2 content) :

- **Un-correlated** content then MPEG-4 data are locally stored on the server,
- **Remotely-correlated** content the MPEG-4 may be transmitted (from a second remote server) over an IP link to the main server,
- **Strongly correlated** content where MPEG-4 data are extracted from the MPEG-2 content present in the incoming transport stream. In that case an MPEG-2 to MPEG-4 trans-coder is required.

2.3.4.   The MPEG-4 Content Encapsulation

Two ways of encapsulation may be considered as specified in [5]. These are described in the following sections.

2.3.4.1.   Simple Encapsulation

In the case where there is no correlation between MPEG-4 contents (no BIFS is required) then a simple encapsulation is required. It consists in the update of the MPEG-2 PSI table to indicate the presence of MPEG-4 data in the transport stream and of the encapsulation in PES packets of the Video and Audio content.

This encapsulation is described on figure 7 below.

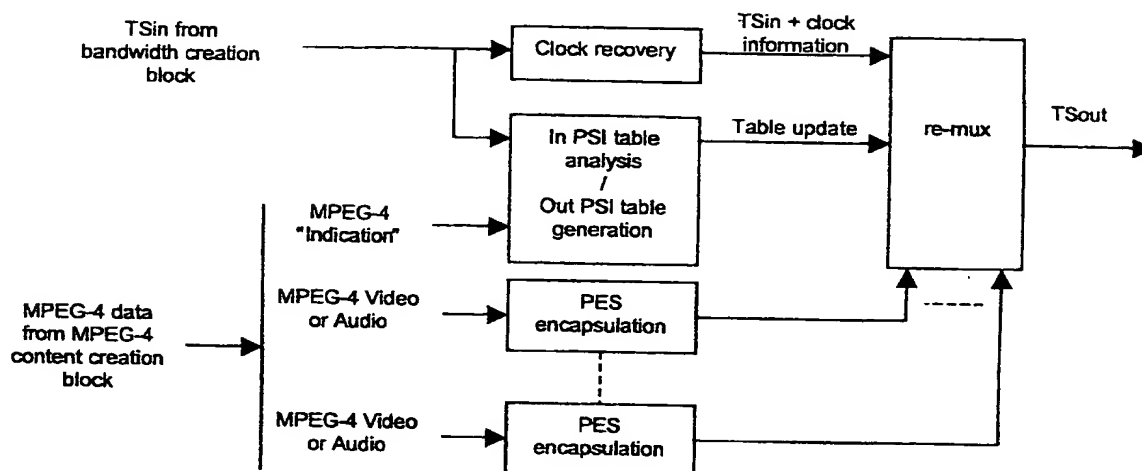| Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

Page 10 / 47

40

Figure 7: Simple MPEG-4 Data Encapsulation (Streams)

2.3.4.2. Complex Encapsulation

In the case where MPEG-4 content is highly correlated (BIFS required) then a more complex encapsulation is necessary. Synchronization Layer and/or the Flex Mux layer of the MPEG-4 standard are then required.

As for the simple encapsulation method the MPEG-2 PSI table have to be updated and supplementary table have to be created to encapsulate BIFS information provided by the Flex Mux and/or Synchronization Layer.

The Audio and Video content has to go through the Flex Mux and/or Synchronization Layer before being encapsulated in the MPEG-2 PES packets. Such an encapsulation is described on Figure 8 below.
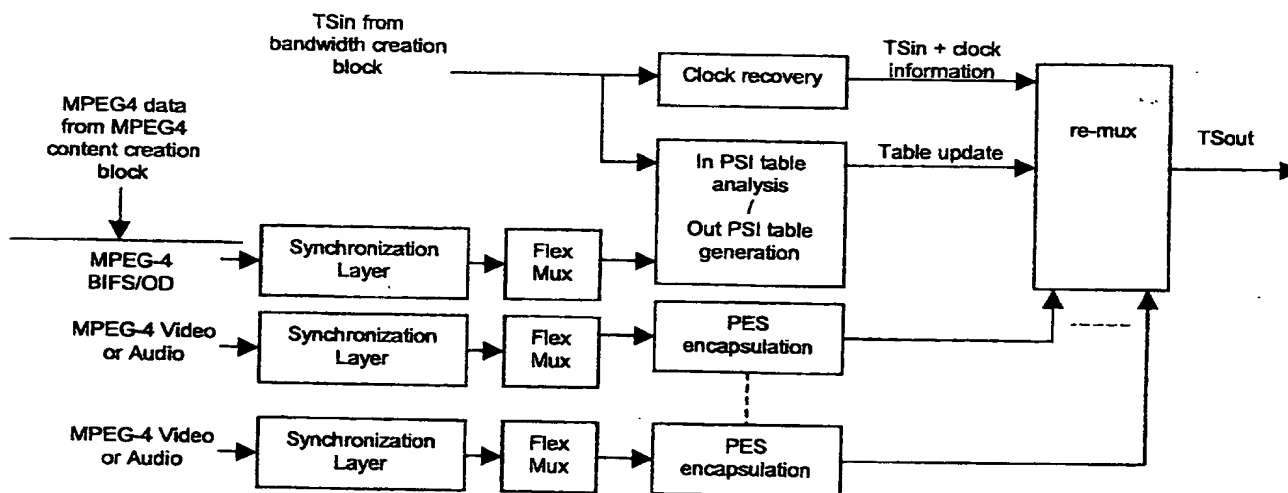
figure 8 : Complex MPEG-4 Data Encapsulation (Scenes)

| Project : | [96397] | Class : | 602 | N° : | [1] | Version : | [1.0] |

40

4-1

It should be noted that for both encapsulations data synchronization is required. In the simple implementation to guarantee that the MPEG-4 data are correctly inserted in the stream and wont lead to a buffer overflow or underflow. This is also the case in the complex implementation. Moreover synchronization may be required at the BIFS level if it is correlated with the incoming MPEG-2 content.

The clock recovery module described in the previous figures is part of the synchronization process.

2.3.4.3.   Server Architecture Summary.

The previous sections have described different implementation of the different block of a MPEG-4 server platform. The different combinations lead to 12 different implementations (Table 1).

	Simple	Complex
Bandwidth Creation	Null Packets Encapsulation	MPEG-2 Bit-rate Trans-coding
MPEG-4 Content Creation	Pre-encoded Content Stored on Disk	MPEG-2 to MPEG-4 Content Trans-coding
Encapsulation Method	MPEG-4 A/V Streams	MPEG-4 Scenes with SL/FM

Table 1 : Server Architecture Summary

2.4. Terminal Architecture

2.4.1.   Terminal Hardware Description

The Terminal is based on PRL Set Top Box (STB) including a TriMedia processor chip and a DIVA MPEG-2 Decoder chip (Rooster base). A view of PRL STB version TM1300 is presented in Figure 4.
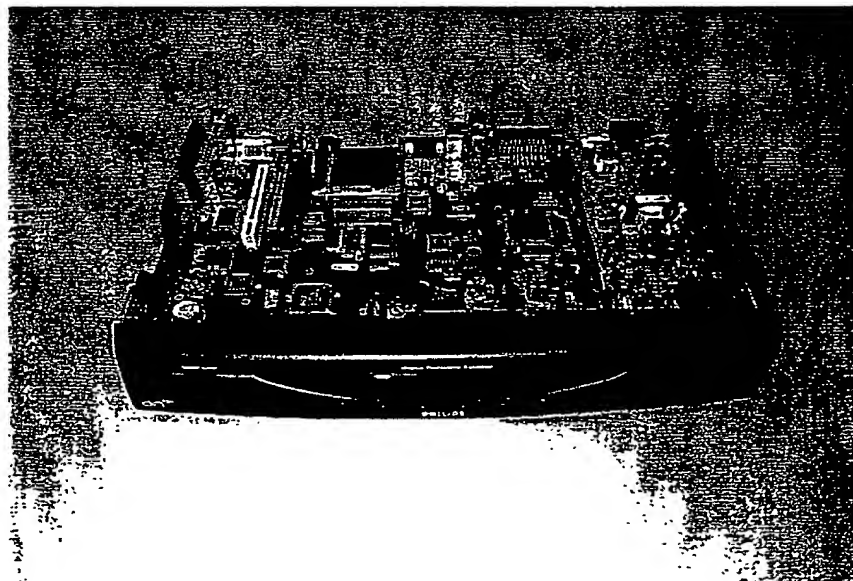

Figure 4 : PRL Set-Top Box

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

42

The STB Hardware specification comprises :

- CPU TM1300 (devices rated at 143MHz)
- 32MB main memory
- Acer M1543 with UltraDMA and USB interfaces
- Battery-backed Realtime clock
- VGA output
- DIVA MPEG-2 A/V decoder
- DVB-S satellite tuner/demodulator integrated on main-board (MPEG2-TS from the tuner front-end can be chained through 2 PCMCIA sockets, supporting Common Interface (CI)
- PSTN front-end on daughter-board
- Full-duplex link controller. isochronous input via TM Video-In port - isochronous output via PCI using DMA

Figure 5 present the STB main functional units.
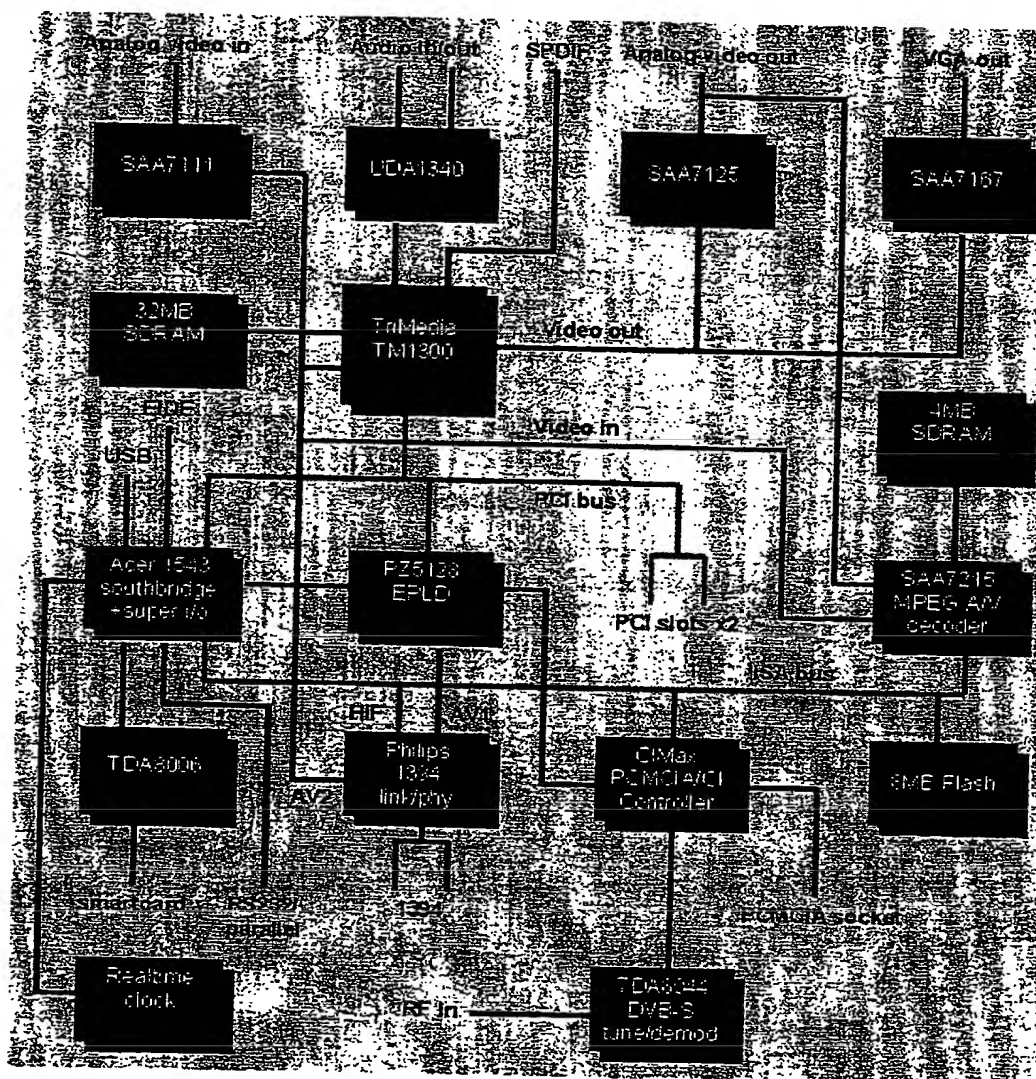
Figure 5 : PRL Set-Top box Functional Units

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

42

4+3

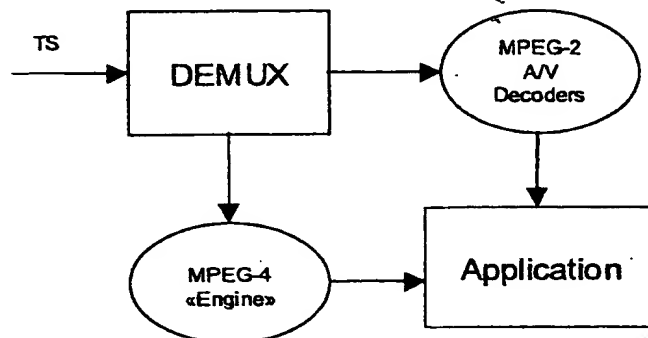## 2.4.2. Terminal Functional Description

Figure 6 : Terminal Functional Diagram

All functions, but the MPEG-2 decoding, are performed in software on the TriMedia.

The de-multiplexer extracts the data from the TS stream, performs the PES des-encapsulation and separates MPEG-2 data from MPEG-4 data. MPEG-2 A/V streams are decoded by the MPEG-2 decoder and the resulting frames are given to the application for rendering.

The MPEG-2 TS demultiplexer has been modified to be able to identify and handle MPEG-4 data. The resulting MPEG-4 data are given to the MPEG-4 "Engine" for further processing (decoding).

2.4.3. The MPEG-4 "Engine" at the Terminal Side

The MPEG-4 engine, at the receiver side, performs the MPEG-4 A/V decoding (Figure 7) and optionally the FM/SL des-encapsulation and the BIFS parser/decoder and the Object Descriptor interpreter, in case of the use of MPEG-4 Systems (for complex scenes, Figure 8).

Figure 7 : MPEG-4 "Engine" for Simple MPEG-4 A/V Streams

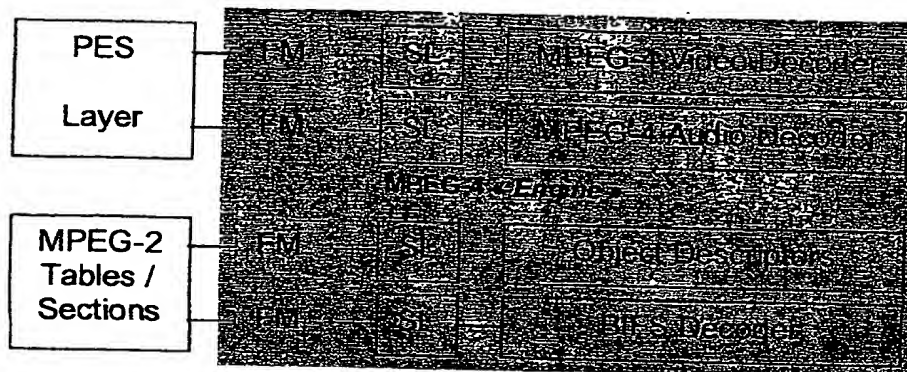| Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

44

Figure 8 : MPEG-4 "Engine" for more Complex MPEG-4 Scenes

2.4.4. Applications

The role of the application is to take MPEG-2 and MPEG-4 A/V streams to build a meaningful output for the end-user.

The foreseen applications concern enhanced digital television combining MPEG-2 A/V streams for a background program and enhanced information provided by MPEG-4 (A/V clips, 2D/3D graphics, etc.).

Concrete examples are listed here and shown on Figure 9 and 10 :

- Interactive Electronic Program Guide (EPG),
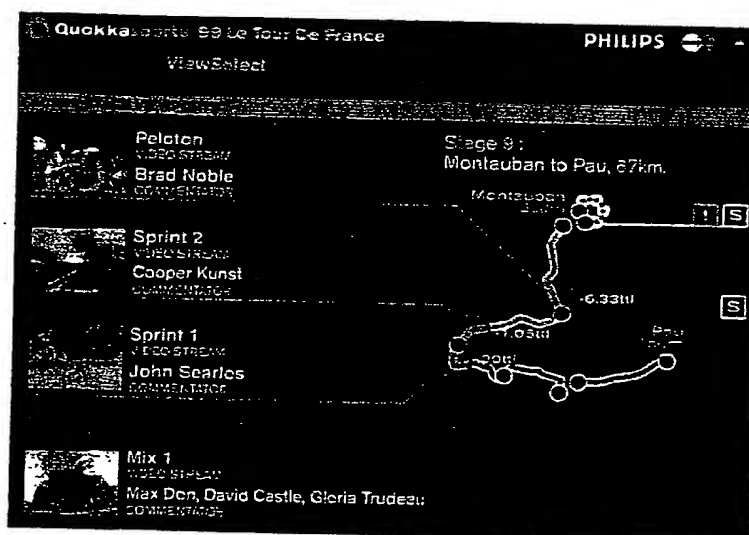- "Immersive Broadcast",
- 7-chanels Formula one.

Figure 9 : "Immersive Broadcast" Application Concept

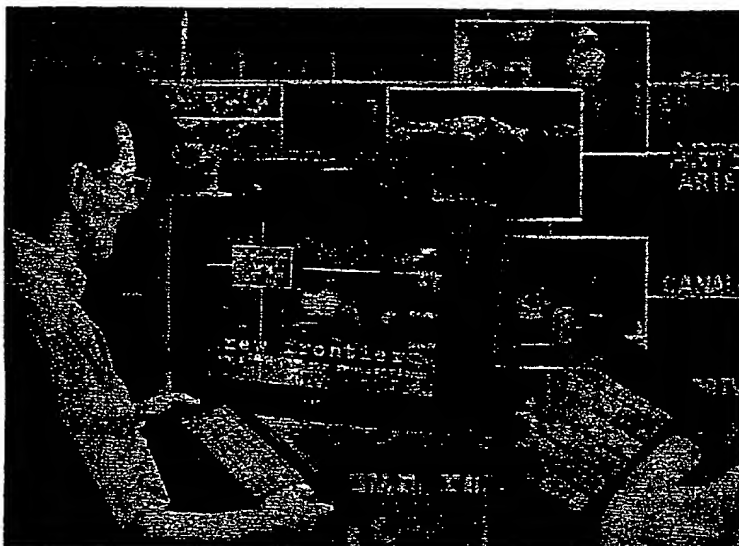| Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

44

45

Figure 10 : MPEG-4 Interactive EPG

2.5. Proposed Set-up Demonstrator
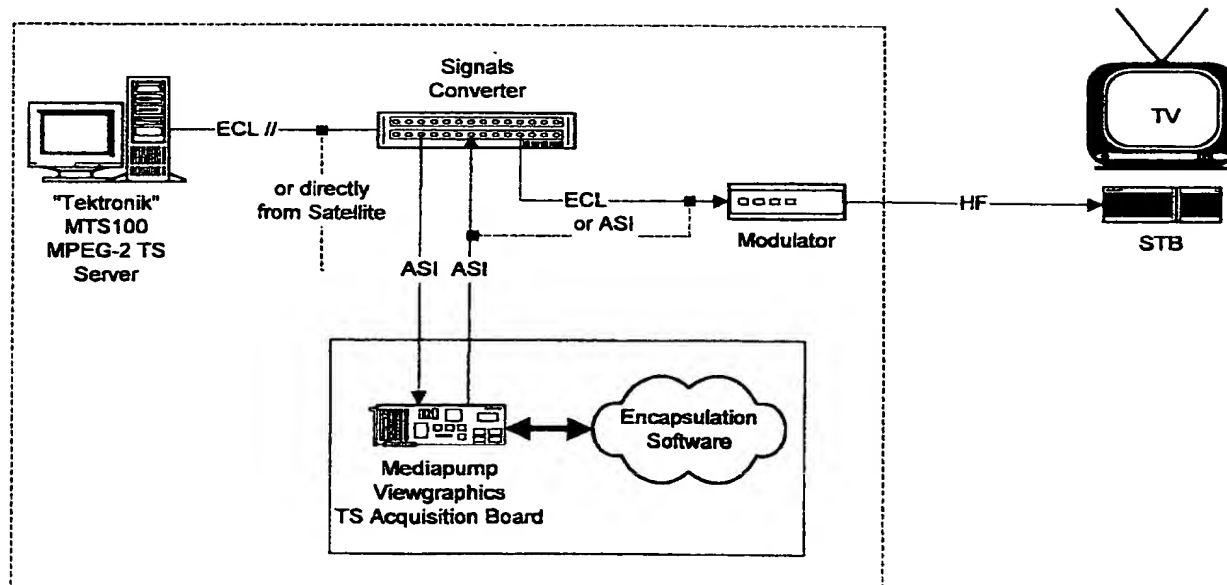
A proposed set-up demonstrator is presented din Figure 11.

Figure 11 : Proposed Set-up Demonstrator

Project :	[96397]	Class :	602	N° :	[1]	Version :	[1.0]

THIS PAGE BLANK (USPTO)

CLAIMS :

1.　　　　　A global end-to-end system architecture for the transport of MPEG-4 data over MPEG-2 Systems, comprising the following parts :

　　　　　(A) a server, for generating an MPEG-2 Transport Stream composed of
MPEG-2 audio/video streams ;

　　　　　(B) a terminal, for retrieving the MPEG-2 and MPEG-4 audio/video data from its input MPEG-2 Transport Stream, in view of a use of said data in a client application ;

　　　　　(C) a link between these two parts.

2.　　　　　In an architecture according to claim 1, a server platform receiving on the one hand an existing MPEG-2 transport stream and on the other hand MPEG-4 data and comprising the following main blocks :

　　　　　(1) a bandwidth creation block, for receiving said incoming MPEG-2 transport stream and creating bandwidth in it in order to insert said MPEG-4 data ;

　　　　　(2) an MPEG-4 content creation block, for receiving said MPEG-4 data and delivering the MPEG-4 content to be encapsulated ;

　　　　　(3) an MPEG-4 content encapsulation block, for encapsulating said MPEG-4 content ;

the output of said MPEG-4 content encapsulation block being the output signal of said server platform, in the form of an MPEG-2 transport stream with MPEG-4 content encapsulated in it ;

　　　　　(4) a fourth block for the control of the operations carried out by the three first ones.

3.　　　　　In a server platform according to claim 2, an encapsulation device consisting of updating means of the MPEG-2 PSI table, in order to indicate the presence of MPEG-4 data in the transport stream and of the encapsulation in PES packets of the video and audio content, and data synchronization means, in order to guarantee that the MPEG-4 data are correctly inserted in the stream, without risk of buffer overflow or underflow.

4.　　　　　In a server platform according to claim 3, an encapsulation device in which the Synchronization layer and the FlexMux layer of the MPEG-4 standard are required and an additional table has to be created n order to encapsulate the BIFS information provided by said FlexMux and Synchronization layers, the audio and video content having then to go through these FlexMux and Synchronization layers before being encapsulated in the MPEG-2 PES packets, and said data synchronization means being also required at the BIFS level.

THIS PAGE BLANK (USPTO)